Structural Processing Methods
for
Speech Signal Analysis

Bhagath Parabattina

146101017

A thesis submitted for the degree of

Doctor of Philosophy (Ph. D.)

Department of Computer Science Engineering
Indian Institute of Technology Guwahati
Assam, India - 781039

June, 2020

October 8, 2020

Dedication

Amma means Love.
I am formed in your womb without my knowledge. You carried me and brought me to this world. You were a good teacher when I was kid, friend when I was young, but you are always a wonderful mom.

Naana means Life
You desire to bring me to this earth, You have been a good friend through out my life. Your philosophy in life gave me an identity in the society. Your teaching is wonderful. After being a father, I can say that
"You are a great father"

Kalyani
When I was alone you were there. My dreams you dreamt, my goals you made me to reach. You managed children nicely in my absence.

It was the plan of God to make all of us family.

Dedicated to my wonderful friend and wife ***Kalyani***
For being amazing friend for past 20 years and ever ending life we are going to have.

It is no surprise why I require God because I know how sinful I am. But it makes me feel curious to know the answer how God could love me even though He knows that I am sinful
- Bhagath

Declaration

I certify that

1. The work contained in this thesis is original, and has been done by myself under the general supervision of my supervisor.

2. The work has not been submitted to any other institute for any degree or diploma

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Guwahati, June 2020

P. Bhagath

Certificate

This is to certify that this thesis entitled **"Structural Processing Methods for Speech Signal Analysis"** being submitted by Parabattina Bhagath to the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, is a record of bona fide research work under my supervision and is worthy of consideration for the award of the degree of Doctor of Philosophy (Ph.D.) of the Institute. The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Guwahati, June 2020

Prof. Pradip K. Das

Acknowledgements

I take this opportunity to thank my supervisor *Prof. Pradip K. Das* for his continuous encouragement, loving-kindness, and guidance during this thesis work. It has been a long association that we have had and there are many inspiring and encouraging moments. The experience with him taught me valuable things like patience and helping towards fellow persons. His cheerful smile makes the environment friendly and affectionate. I had a desire in my Ph.D duration to reach lab before he comes to the Department. Sometimes, I succeeded but failed many times. Anyhow, it didn't have any influence on his response.

It also gives me pleasure to convey my gratitude to the doctoral committee members Prof. Shivashankar B. Nair, Dr. Pinaki Mitra, and Dr. Rashmi Dutta Baruah for continuously advising me throughout the duration of the Ph.D. program. I never felt any professional pressure in our conversations. This credit goes to them for the cordial relation that they maintain. I have a long association with Nair Sir as I have with Das Sir. He is another person with a peaceful smile that gives a warm welcome. I would like to thank all the faculty members and technical staff of the Department of CSE. Their friendly approach towards students is appreciable.

When I think about writing this part, it melts my heart when I remember my friends for their support and friendship. I am very much blessed with wonderful people in my life. I thank all friends who were supportive during my needful times. I thank my childhood friends Mr. Nirmal Kumar and Dr. Prasad, Vishal Parage, Dr. Shrishendu Das and Sathish for their long and adorable friendship.

I especially thank Deepak Raj, Sandeep Vidyapu, Dr. Mohit, Dr. Sonia and Dr. Tushar for the moments that we spent. There would be no progress in research if there is no professional environment, encouragement and good people around us. I thank my co-researchers Prasanta Roy, Deepankar Nankani, Pallabi Saikia, Vanshali Sharma, Megha Jain

and Subham Jain for the valuable discussions. Sandeep is the only person in the department who visited me regularly and we are like chai friends. Deepak is the friend that I met at the beginning of Ph.D and he has been a good friend of mine for these six long years. Deepankar is another cheerful guy in the lab with a peculiar smile. He and Pallabi are very friendly people. Prasanta and I shared good moments having midnight coffee around the philosopher's table. It was a great moment that I spent at the end of my PhD. Vanshali became a good friend in a short time. I should mention the members of the Speech lab Susma and Komal also for their jolly presence.

I thank all the students of B.Tech and M.Tech who traveled with me for the duration of my Ph.D. program. I was encouraged by all the thoughts and energy they showed while solving the problems.

There are many co-research scholars who were associated with me for six long years by whom I was motivated. I cannot stop mentioning few friends with whom I spent this period. Among them, Mr. Hema, Mr. Surajit, Mr. Swaroop, Mr. Akash Anil are few people who smiles at me and spent a moment when we meet. These small moments are also important in a scholar's life. The time I spent with my juniors was not long time. But there are some cheerful moments that we spent and they are part of memories. Miss Divya, Mr. Suraj, Ms. Menaxi are among them. This list cannot be exhausted but cannot be paused. There are friends with whom I spent different occasions very cheerfully and not related to academics. I thank Prabhakar, Chiranjeevi, Badal Soni for the moments that I spent with them in campus.

During the days that I spent, I was associated with a church in North Guwahati where I was spiritually strengthened. The members of the church prayed for me and my family when I was in trouble. I thank each and every one there, namely Bro. Roy and family, Bro. Prakash and family, Bro. Sanjog and family, Bro. Sunil and family, Bro. Venkat and family, Bro. Santhosh and family, Bro. Joseph and family, Mrs. Anitha, Ms. Feba, and few youngsters. Anitha is like my younger sister. Dr. Tarun and Bro. Sunil are good friends with whom I had a great time. I enjoyed the presence and greetings from Dr. Lyngdoh, Mrs.

Lyngdoh, and Dr. Nelson and sister. I should mention about two people who became tools in hand of God for bringing good news into my life. I thank Dr. Rajendra and uncle Thambi for their love and approach.

It will not be complete if I don't thank my parents Mr. Peter and Mrs. Vasantha without whom I am nothing in this life. This gives me immense pleasure to thank them for giving this life, showing patience, spiritual guidance and continual love. The painful moments that they bear and long watchfulness brought me up. I thank my aunt Mrs. Jhansi and uncle, Mr. Prasad, for they are like my second parents and showed wonderful love to me. I also thank my grandparents for their unconditional love. I thank my aunt Rachel and sister Jones (of course my brother-in-law) for their spiritual enlightenment and prayers. I take this moment to thank all my teachers who were part of my life in building up my scientific mindset, moral values and dedicated teaching. Especially I thank my teacher Mr. Rambabu who taught me Mathematics during my 10^{th} standard which helped me to understand the fundamentals. I also thank my childhood teachers Julius and Ezra for their loving guidance.

I especially want to mention an important episode during the lock down period due to COVID. I stayed in campus during this period with the help of Deepak, Dr. Prakash and Mr and Mrs. Dharma Lingam. I thank these people especially for their friendship and extending the hospitality. I also thank Mrs. Das and Mrs. Nair for the hospitality and being my well wishers. Lastly I must thank my son *Akshay* and daughter *Akhila* for being an inspiration during the last days to finish the thesis. The longing to meet them drove me to complete the work.

Guwahati, June 2020

P. Bhagath

Contents

List of Figures

List of Tables

Abstract

Speech signal analysis is a crucial study that helps to develop methods for problems like phoneme segmentation, speech recognition, speaker verification, etc. There are various frameworks and techniques that support these problems. Frameworks like Hidden Markov Modeling and Deep Learning are popular. The frameworks are efficient with large data sets where intensive training is possible. However, this becomes challenging in case of under-resourced language since sufficient data cannot be provided for the intensive training.

To address the needs of these languages, suitable methods are required with the capability to seek for significant clues with less amount of data. Structural processing methods focus on understanding the signals differently compared to signal processing methods. In this approach, a signal is treated as an image rather that a time series with different samples at different time stamps. The need for these methods arises due to the limitations in Hidden Markov Models. HMM contains states in which each state depends on at most two neighboring states. This limits HMM to have a holistic view of the entire signal.

Recent developments in graph signal processing techniques give a way to analyze the signals by using graph data structures. These methods enable to use combination of temporal relations and frequency components while modeling the signals. The thesis addresses the problems of speech characterization and segmentation while considering the above mentioned issues. Different features like trajectories and Tree structures are proposed and found to be useful for modeling speech signals that can be used further for recognition. Three different features based on trajectories, graph structures and fractals are proposed for segmentation task. The experiments were conducted on Indian accented spoken English vowels, words and TIMIT sentence data. Tree structures and trajectories were found to be useful in characterizing vowels and words, respectively. In the phoneme segmentation experiments, words data were collected from people belonging to different regions of India. The segmentation approaches are ascertained to be appropriate for finding phoneme boundaries of phonetic units in spoken words and sentences. The algorithms and obtained results are discussed in the thesis.

Publications

1. **P. Bhagath**, P K Das, "Phoneme Boundary Detection using Graph Structures and Graph Eigen values,"International Journal of Pattern Recognition and Artificial Intelligence, World Scientific (Under preparation).
2. **P. Bhagath**, P K Das, "Detecting Phonetic Boundary Points using Fractal Analysis and CCA," (Under preparation).
3. **P. Bhagath**, P K Das, "A CCA based Phoneme Segmentation using Quadrilateral Parameters of Speech Trajectory," (Under preparation).
4. **P. Bhagath**, P K Das, "Phoneme Boundary Analysis using Multiway Geometric Properties of Waveform trajectories towards Low Resource Languages," 1st Joint SLTU and CCURL Workshop, Marseille, FRANCE, 2020, (SLTU-CCURL 2020), Pages 144-152.
   Conference Link: http://sltu-ccurl-2020.ilc.cnr.it/
   Paper: https://www.aclweb.org/anthology/2020.sltu-1.20.pdf
5. **P. Bhagath**, Megha Jain and P K Das, "Dynamic Speech Trajectory based Parameters for Low Resource Languages," MIND-2020 - 2nd International conference on Machine learning, Image processing, Network security and Data Mining, NIT Silchar, India, 2020 (Accepted).
   Conference Link:http://mind2020.nits.ac.in/
6. **P. Bhagath** and P. K. Das, "Characterization of Spoken English Vowels using Tree Structures," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 1758-1763.
   Conference Link: https://www.tencon2019.org/
   Paper Link: https://ieeexplore.ieee.org/document/8929557
7. **P. Bhagath** and P. K. Das, "Phoneme Boundary Analysis Using Graphs," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 1764-1768.
   Conference Link: https://www.tencon2019.org/
   Paper Link:https://ieeexplore.ieee.org/document/8929673
8. **P. Bhagath** and P. K. Das, "Phoneme Segmentation using Fractal Analysis," OCOCOSDA 2019 - The 22nd Conference of the Oriental COCOSDA, Cebu City, Philippines 2019.
   Conference Link: http://www.orientalcocosda2019.org.ph/
9. **P. Bhagath** and P. K. Das, "Acoustic Phonetic Approach for Speech Recognition: A Review," NSA 2016 - International Symposium on Acoustics, KIIT Gurgaon, India 2016.
   Paper Link: https://www.researchgate.net/publication/313102457_Acoustic_Phonetic_Approach_for_Speech_Recognition_A_Review

Acronyms

HMM	Hidden Markov Model
DNN	Deep Neural Network
LPCC	Linear Predictive Cepstral Coefficient
MFCC	Mel Frequency Cepstral Coefficient
SOFM	Self Organizing Feature map
STM	Stochastic Trajectory Models
SFHMM	Segmental Feature Hidden Markov Model
PSMs	Polynomial trajectory Segmental Models
MMSE	Minimum Mean Square Error
MD	Merge Distance
MDS	Multi Dimensional Scaling
DB-SCAN	Density based spatial clustering of applications with noise
MMTD	Maximum-Minimum Trajectory Distance
STD	Semantic Trajectory Distance
G-HMM	Gaussian Hidden Markov Model
GSP	Graph Signal Processing
ECG	Electro Cardio Gram
TTS	Text To speech Synthesis
DC	Direct Current
ZCR	Zero Crossing Rate
HDMs	Hidden Dynamic Models
CCA	Canonical Correlation Analysis
LTSM	Linear Trajectory Segmental Model
SNR	Signal to Noise Ratio
K-NN	N- Nearest Neighbors
CUSUM	Cumulative Sum

Chapter 1

Introduction

Speech recognition is the process of understanding and extracting information in speech signals. It has multiple applications in various domains like service industry, IoT (Internet of Things), mobile devices, etc. It has got significance in the service industry where an organization provides services through voice-enabled machines. IoT domain combines devices that can handle different kinds of activities and gives accessing to these operations targeting towards a particular task. Recent developments have increased the capabilities of hand-held devices which opens a room for speech processing tools to improve the services. Examples for this kind of applications are Google Voice Assistant, Apple Siri, Microsoft Cortana, Amazon Alexa and so on. Apart from the commercial products, there are many other platforms that support developers to build tailor-made systems. Researchers are working to develop methods and procedures that can improve the recognition process. There are well known frameworks in this area such as Hidden Markov Models (HMMs), Neural Networks, Deep learning, etc. The common steps in any framework are as follows:

1. Language Modeling
2. Data acquisition
3. Feature Extraction
4. Feature Modeling

The first step is specific towards languages where the characteristics of the language is well understood to build a common structure for components of a language. The second step gathers the speech data as recorded signals and the main characteristics of these signals are transformed as features. These features are to be modeled consequently to build systems that can recognize live data. There are different techniques and tools that have been used at different levels of the framework. Feature extraction techniques play a crucial role in processing and understanding the speech signals. In general, this step can be handled in number of ways depending on the signal representation. There are two ways of signal representation known as temporal representation and spectral representation. The first one emphasizes on the temporal dynamics of a signal whereas the second concentrates on the nature of frequency components. There are different techniques for feature representation such as Linear Predictive Cepstral Coefficients (LPCCs) method [1], Mel Frequency Cepstral Coefficients (MFCCs) [2] that are successful in many speech- based tools. These features are modeled using methods like Markov models, Neural Networks, Deep Learning, etc. They are efficient in modeling the speech signals for they use large amount of data sets. Even though these methods are effective, research has been done to address issues in specific language contexts. This is important because of less availability of data sets for some languages that are known as low resource languages. The challenge is to solve the problem of speech recognition with small amount of data. The success of this task depends on the efficiency of the steps that are discussed earlier. The requirement of huge data sets is due to the fact that the methods do not concentrate on useful and simple clues of speech signals. One of the approaches that helps to understand the clues so that modeling can reflect these dynamics is structural characteristics of signals.

1.1 Motivation

The motivation for structural processing of speech signals has been addressed in [3] which discusses that conventional HMMs could not model effectively the time dependency among the frames in a speech signal. The effect of this is neglecting essential properties of speech dynamics. Another problem is that dynamic articulation differences are not considered in the modeling stage. To overcome these problems, frameworks that work beyond the nature of HMMs have to be used for acoustic modeling. Another crucial thing that is to be considered is that phonological structures of spoken units can contribute to speech processing systems by adding important clues. Each phonetic unit can be described by its unique structure. Structural distinctiveness is helpful in many problems related to speech signal analysis [4]. Each speech activity occurs with the vibrations of vocal cords. The generated speech signal is influenced by the physiological characteristics of the vocal cords and the passage in which it travels. The acoustic signal that is produced has a distinct structure for different sounds. If the speech activity is considered as an event, it makes a path that the object travels forms a different structure for each sound. The phonological structures of spoken units can be modeled in different ways. Acoustic phonetics deals with modeling of speech signals. The characteristics of phonetic units can be well understood from the geometrical properties of speech waveforms also. These properties can give the clues that are important to understand and represent features of speech sounds. Perceptual constancy [5] says that human ear can perceive the sounds that are similar even though they have small variations in its structure. In the present work, the structural components are used to define an appropriate representation so that features of temporal dynamics [6] are used to model the underlying speech signals. We proposed methods that can represent speech signals and are useful for processing them. The contributions in this thesis are discussed in Section 1.3.

1.2 Brief literature review

A trajectory is the path that an object in motion follows through space as a function of time. Any object in motion through space may be called a Projectile. Yifan Gong [7] defined trajectory in speech signal as a sequence of moving points. H. Gish and K. Ng [8] proposed a segmental speech model for spotting keywords. The proposed segmental model represents a speech segment by a set of features which includes a time varying trajectory, a residual error covariance around the trajectory and the number of frames in the segment. This feature set works as a model for a segment of speech signal. The segmental model can be represented as:

$$C = ZB + E \tag{1.1}$$

C in Equation 1.1 is calculated for N frames where each frame is represented by a D dimensional feature vector. B is the parameter matrix for trajectory and E is the residual error matrix. When combining models of different segments of a speech signal, normalization is applied to maintain uniformity among the segments. This normalization is handled by matrix Z. The parameters in this trajectory model are estimated by Maximum Likelihood estimation procedure. This procedure uses likelihood of a segment and probability of the model. These parameters are re-estimated so that the final parameters are computed for the given segment. The procedure proposed in [8] was used to classify vowels and secondary processing algorithm for spotting keywords. The primary model was obtained by Hidden Markov Modeling (HMM). The concept of parametric trajectory model has been extended to include time varying co-variances in [9]. This allows to observe changes of co-variance structures along the trajectory. The authors also proposed a method for distance measurement between speech segments based on trajectory models. This was achieved by allowing three different co-variance matrices existing over a single segment. B in Equation 1.1 is matrix of order $R \times D$, where R is the number of parameters used in the trajectory model and D is the number of dimensions. The nature of parameters may be constant, linear or quadratic based on the value used in R. It is proved in the parametric trajectory models that the performance is superior using quadratic models.

This model has been investigated further in [10] and found to be useful for large vocabulary speech. The authors developed Bayesian adaptation method for polynomial trajectory segment model. In this method, Bayesian approach has been employed to estimate the parameters. Instead of direct estimation, a shift in the parameters is used which is useful in sharing these parameters across a class of models. We found a variation of this approach which combines the Bayesian approach and neural networks in [11]. Here the authors investigated trajectories of speech signal in Self Organizing Feature Maps (SOFMs). SOFM is a neural network which has the capability to preserve the topological relationship of the input space. But the challenge here is to represent multi-dimensional speech signal on a 2-Dimensional network. To address this issue, posterior probability of the response is maximized to obtain a more reliable trajectory from SOFM. The obtained trajectory is like a graph in the SOFM network.

Apart from the direct approach of considering trajectory as it is, researchers tried to use context information in the trajectory for different acoustic context. This kind of approach is found in [7]. Here, context and duration of trajectories have been integrated in the modeling. The procedure was motivated by understanding the need of including context of the trajectory in a speech signal. The trajectories that are related to a particular acoustic context can be clustered and can be used further to represent context variability also. Later on the focus has been shifted to statistical modeling [12]. This modeling technique assumes the feature vector is a point on mean path which has a number of straight line segments. The model tries to understand the amount and rate of deviation of a trajectory from this mean path.The parameters are estimated by using the Expectation Maximization algorithm. One of the drawbacks in Hidden Markov modeling is trajectory folding i.e. HMM cannot discriminate the context from which a particular phoneme has arrived. It is because of the limitation of the first Markov process where the past observations do not influence the future observations. There are many ways in which this problem can be solved. One of the solutions can be Stochastic trajectory models (STM) [7]. In this method, clusters of trajectories have been modeled by a mixture of probability density functions. In [13], we can find linear trajectory segmental model in which the trajectory is defined with two parameters slope(m) and mid-time value(c). These two parameters are used to characterize segmental behavior. The distribution of intra segment and extra segment are assumed to be gaussian in this model. The model parameters can be found by differentiating with respect to m and c. Segmental Feature Hidden Markov Model (SFHMM) [14] was proposed to overcome weakness of the observation independence asumption in conventional HMM. Parametric trajectories were used as features in this modeling technique. The features of each segment contain acoustic context information of adjacent segments. To adjust frames of different lengths, time-normalization have been employed. Like conventional HMM, the parameter re-estimation have been done using methods similar to Baum-welch method.

One issue in segmental modeling methods is it’s usage of HMM recognition algorithms. This didn’t help the methods to be established as complete alternatives to conventional HMMs. The work reported in [15] proves that the understanding of relationships between cepstrum, delta-cepstrum and delta-delta cepstrum helps to combine the trajectory methods and HMM. In this technique, the trajectories are used for deciding the state sequence in Viterbi algorithm. This could avoid discontinuity in mean squares obtained in HMM procedure. The extension of this work can be found in [16]. The method proposed here can work on HMMs with multiple Gaussian distributions. While selecting the state sequence in Viterbi decoding, this method allows to select the state with the best Gaussian distribution among the available states.

1.3 Contributions

This thesis addresses the above mentioned issues through structural methods and these methods are proved with application in recognition and segmentation tasks. The contribution of the thesis is to understand the geometrical properties of speech signals and find the suitable applications of these properties so that the objectives of structural properties are served. The following list summarizes the contributions:

1. Trajectory features
2. Graph based methods
3. Fractal methods

1.4 Organization of the thesis

The thesis is organized as follows:

1. Chapter 2 discusses properties of the trajectory parameters that are useful for studying signal characteristics. These properties are studied in the context of Indian accented English spoken vowels. The chapter provides the study of two classes of features in detail.
2. Chapter 3 gives a new representation to trajectory properties. The method that is discussed is known as Tree based structural analysis. This approach proposes a new arrangement of data components to process and characterize the spoken units.
3. Chapter 4 explains geometrical properties for another problem i.e. Speech segmentation. This work proves that the properties studied for characterization or recognition can also be useful for finding the boundaries of different spoken units present in a speech signal.
4. Chapter 5 provides an advanced method based on GSP to address the problem of phoneme segmentation. This method is distinct in handling the attributes of waveforms to make useful for finding the boundaries.
5. Chapter 6 describes a fractal based approach for phoneme segmentation.
6. The concluding remarks of the thesis and future perspectives are discussed in Chapter 7.

Chapter 2

Speech Signal analysis using Trajectory Parameters

This chapter describes the importance of trajectory analysis for speech signal characterization. Here, we propose two classes of features where the dynamic nature of the signals are captured as trajectory components. In Section 2.1, the significance of the problem and solution with trajectory parameters is discussed. A brief review on trajectory and similarity measures are discussed in Sections 2.2 and 2.3, respectively. The proposed method with the feature extraction method is explained in Section 2.4. The data set used and the environment of the program development are detailed in Section 2.6 and the results are elaborated in Section 2.7. Finally, the future work is described in Section 2.8

2.1 Introduction

Trajectory is a path followed by an object with a proper direction. A speech signal can be treated as a trajectory which is influenced by a particular speech activity. Each such activity records the events in different styles and consists of distinct structures. The structural components that are available in this path can be used to model the speech events appropriately. Trajectory modeling helps to incorporate the temporal dynamics of phonetic units. These include the changes and variations in structure of different phonetic sounds. This helps in syllable classification that uses linguistic features such as syllable duration, lexical stress and difference between mono and poly-syllabic words [17].

The main advantage of trajectory with the combination of HMM has been addressed by many researchers and is widely accepted technique. But the issue with the existing mechanisms is its computational efficiency. The cost reduction is important because it can affect the performance of the overall system. This requirement may not be crucial for high configuration computers, but it matters for low computational devices. The present feature extraction methods are targeted towards less-expensive systems. The first benefit of the parameters is reduction in the complexity of feature computation. Second is the comparatively less space requirement for the features than the popular methods like LPCCs and MFCCs. As a result, the training time required would be less. Parameter extraction methods are discussed in detail with implementation technicalities. To prove the effectiveness of these features, HMM was used for modeling. It is found in the study that the proposed parameters are effective for the speech classification problem.

The main contribution of the work is proposed in [18]. The proposed features can be classified into two categories as follows:

1. Peak attributes
2. Fréchet distance based parameters for waveform trajectories

2.2 Related Work

In a trajectory model, a speech signal is represented using parametric trajectory models given by Equation 2.1:

$$C(n) = \mu(n) + e(n), \forall n \in \{1, ..., N\} \tag{2.1}$$

where $C(n)$ is the set of cepstral properties in a speech segment of length N, $\mu(n)$ is the mean feature vector and $e(n)$ is the residual error term. H. Gish proposed a trajectory model for vowel classification that uses Gaussian Mixture Models and time-varying covariances [8]. Another variant of trajectory models is Polynomial trajectory Segmental Models (PSMs) that can be used for modeling co-articulation effects through context-dependent models. The PSM systems assume that the observations are generated by a Gaussian process and the co-variance is assumed to be constant over a segment. The basic parameter that is used here is a time-varying vector mean trajectory and is expressed in Equation 2.2:

$$\mu(t) = b_1 + b_2 t + ... + b_r t^{r-1}, \forall t \in [0, 1] \tag{2.2}$$

where t is the normalized time [19]. Even though HMMs are successful, they use less knowledge of the underlying signal. HMM associates each state with a single frame of the speech signal which doesn't capture the intra-segmental temporal variations [13]. Therefore, an alternative process called Segmental HMMs has been used to model speech signals using parametric trajectories. In this process, a trajectory is obtained by using a design matrix based on transitional information of contiguous frames. The model of this system can be expressed using Equation 2.3:

$$P(C_t|S_i, \lambda) = P(ZB_t|S_i, \lambda)P(C_t|ZB_t, S_i, \lambda) \tag{2.3}$$

where C_t is the observation vector, ZB_t is the unique trajectory at time t, λ is the observation probability of C_t that occurs at state S_i [14]. M. Firouzmand proposed a discrete cosine model for amplitude trajectories of the form given by Equation 2.4. Models are estimated using the amplitudes of trajectory using Minimum Mean Square Error (MMSE) [20].

$$A_i(n) = \sqrt{\frac{2}{N}} \sum_{p=0}^{p_i} A_{ip} W(p) cos\Big((n + \frac{1}{2}) \frac{p\pi}{N} \Big) \tag{2.4}$$

Modeling continuous signal is beneficial to capture the dynamic nature of the entire signal. In the present approach, features are extracted over the entire signal where intra-segmental properties are captured effectively. In the next subsection, different works related to trajectories in finding similarities are described.

2.3 Trajectories in similarity analysis

A trajectory can represent spatiality and order of the data with respect to time. Analyzing trajectories can help to classify similar entities based on the relationships found. They are useful in various applications that include GPS data, user profiling, location prediction, time series analysis, pattern mining, etc. Methods that are useful in finding the patterns are listed as follows:

- Merge Distance (MD)
- Multi Dimensional Scaling (MDS)
- Density based spatial clustering of applications with noise (DBSCAN)

Zelei et al. proposed a similarity finding method for predicting location based on a person's mobility features. This method is intended to find relationships between social relations of a person and variances in trajectory so that the moving location can be predicted apriori [21]. There have been applications in which spatial and temporal features alone cannot give sufficient information about system behavior. This requirement has lead to the use of multiple features based trajectories. In these approaches, a trajectory is represented through a combination of three or more features. These methods are called data fusion techniques. They merge the dynamic nature of different similarity properties to generate a model [22]. The model used in this approach is given in Equation 2.5:

$$MMTD(t_1, t_2) = 1 - (w_1, w_2) \begin{pmatrix} dist_1(t_1, t_2) \\ dist_2(t_1, t_2) \end{pmatrix} \tag{2.5}$$

where $dist_1$ and $dist_2$ are different similarity measurements and each measure is treated with unequal weightages. MMTD is a maximum-minimum trajectory distance.

Zedong et al. proposed a method for predicting location based on user similarity that uses GPS trajectories. This approach combines spatio-temporal features and GPS coordinates data to extract the similarities among different users. The ordering of the points can be achieved by incorporating timestamps into the system. To understand the similarity, a Semantic Trajectory Distance (STD) was used. This distance is given by Equation 2.6.

$$STD = 1 - \frac{|lcs(T_1, T_2)|}{|T_1| + |T_2| - |lcs(T_1, T_2)|} \tag{2.6}$$

where $|T_1|$ and $|T_2|$ represent the length of trajectories and $lcs(T_1, T_2)$ is a measurement used to define the longest common sub-sequence. This method was proven to be effective in finding the similarity between user mobility [23].

Trajectory data analysis has been used to improve navigation systems and traffic management also. Here, the navigation path was represented with features like traffic flow information, location and motion. To build rich navigation systems, clustering algorithms that are formed by a number of techniques have been used. One crucial similarity measurement used in such clustering algorithms is Merge Distance (MD) [24]. For a trajectory T that consists of a sequence of points where each point is represented by a time point and distance between these points is given by $l(p) = \sum d(p_i, p_j)$, the Merge distance is the length of the shortest trajectory that can represent two different trajectories and is given by Equation 2.7:

$$MD(t_i, t_j) = \frac{2l(t_i, t_j)}{l(t_i) + l(t_j)} \tag{2.7}$$

The present work focuses on proposing spatio-temporal features for speech trajectory analysis. Shape-based signal properties are proposed for characterizing speech signals. In the next section, the features and methodology are discussed in detail.

2.4 Proposed trajectory features

The present approach assumes that a signal is a series of segments where each segment is a sequence of points. To characterize and understand similarity patterns in a signal, two different parameters are defined as follows:

1. Peak attributes
2. Similarity distance measures

The common nature of these features is to capture the dynamic structural changes in the entire signal as components that contain the crucial phonetic characteristics. To do this, the proposed features concentrate on using the shape of the signal to model the temporal patterns. Each subsequent subsection gives a detailed explanation of these features.

2.4.1 Peak attributes of speech trajectory

These features focus on the dynamic nature of a signal in terms of signal's spatio-temporal behavior. The shape of the signal is characterized with a set of primitives. They are listed as follows:

- Peak
- Valley
- Peak width

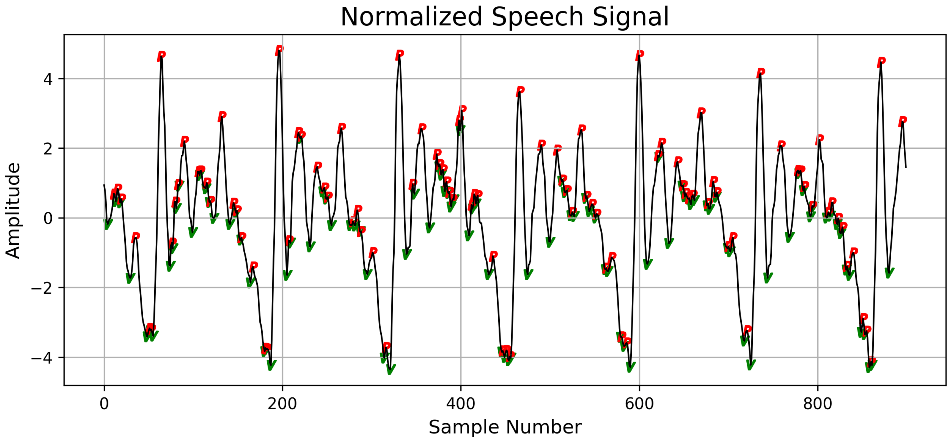


Figure 2.1: Peaks and valleys in a speech segment of vowel /a/

In a segment of speech let s_{i-1}, s_i and s_{i+1} be consecutive samples, the terms mentioned above are defined respectively as follows:

Definition 1 s_i is said to be a peak if $s_{i-1} < s_i > s_{i+1}, \forall i \in \mathbb{Z}$

Definition 2 s_i is said to be a valley if $s_{i-1} > s_i < s_{i+1}, \forall i \in \mathbb{Z}$

Definition 3 The sample p_k being a peak point between any two valleys v_q and v_r, the difference $|r - q|$ is defined as width for the peak p_k $\forall k, q, r \in \mathbb{Z}$ and $q < k < r$.

To understand the concepts, let us consider a sample speech segment shown in Figure 2.1. A peak is a local maxima in the signal whereas local minima is a valley. The peaks and valleys are colored in red and green respectively. The central idea of the present approach is that a speech signal is treated as a trajectory that records different acoustic events at various instances of time. The properties of these events are analyzed to find similarities among them. They are further modeled to form a generic representation for these trajectories. Peaks are considered as acoustic events and their attributes are used to understand temporal variations in a signal. When a moving object is observed in terms of the path that it forms in a trajectory, peak width is the duration that an object spends in a particular event. With the changes in its duration, a new peak forms in the path of a trajectory. The duration varies for each event by which a pattern can be observed for an object that can distinguish among different phonetic structures. Since each phonetic unit has a unique structure, the path that the spoken units can form is also distinct in its structure [4]. In the present study, this dynamic nature of the spoken units is taken and the similarity between them is used for classifying them. The classification is achieved by using Hidden Markov Modeling (HMM). The steps involved in parameter extraction are given in Algorithm 1. The peak widths of vowels /a/, /e/, /i/, /o/ and /u/ are shown in Figure 2.2. The second proposed feature extraction method is explained in the next subsection.

Algorithm 1: Trajectory parameter extraction

```
Input:
$S_n$: Input speech signal
Output:
$T_n$: Trajectory vector that contains peak attributes
begin
    for $i \leftarrow 0$ to $length(S_n)$ do
        if $s_{i-1} > s_i < s_{i+1}$ then
            $V_i \bigcup$ i                    ▷ Find valley positions for the given signal
    for $j \leftarrow 0$ to $length(V_n)$ do
        $T_i \bigcup |V_{j+1} - V_j|$          ▷ Find the peak widths
    return $T_n$
```

2.4.2 Fréchet distance based curve parameters of speech trajectory

The proposed features give an insight into the dynamic nature of a signal which captures structural changes over the segments. Therefore, the variations of the entire signal can be reflected in the features. Usually, features are dependent on the motion of trajectory that varies at different instances of time. The normalized signal is divided into a number of frames and the dynamics of speech signal are represented as structural changes in the trajectory curve. Each trajectory represents the motion of an object across different time instances. The dynamics or structural changes between adjacent frames are represented by understanding the differences between the motion of trajectory. These properties are represented by Fréchet distance between two curves. Each speech signal consists of various acoustic events where those events can be characterized by the vibration of the vocal tract. This is distinct for different events.

Fréchet distance [25] is a measure of similarity between curves which preserves the order of data along with a time series. Let τ_1 and τ_2 be two trajectories that represent paths of

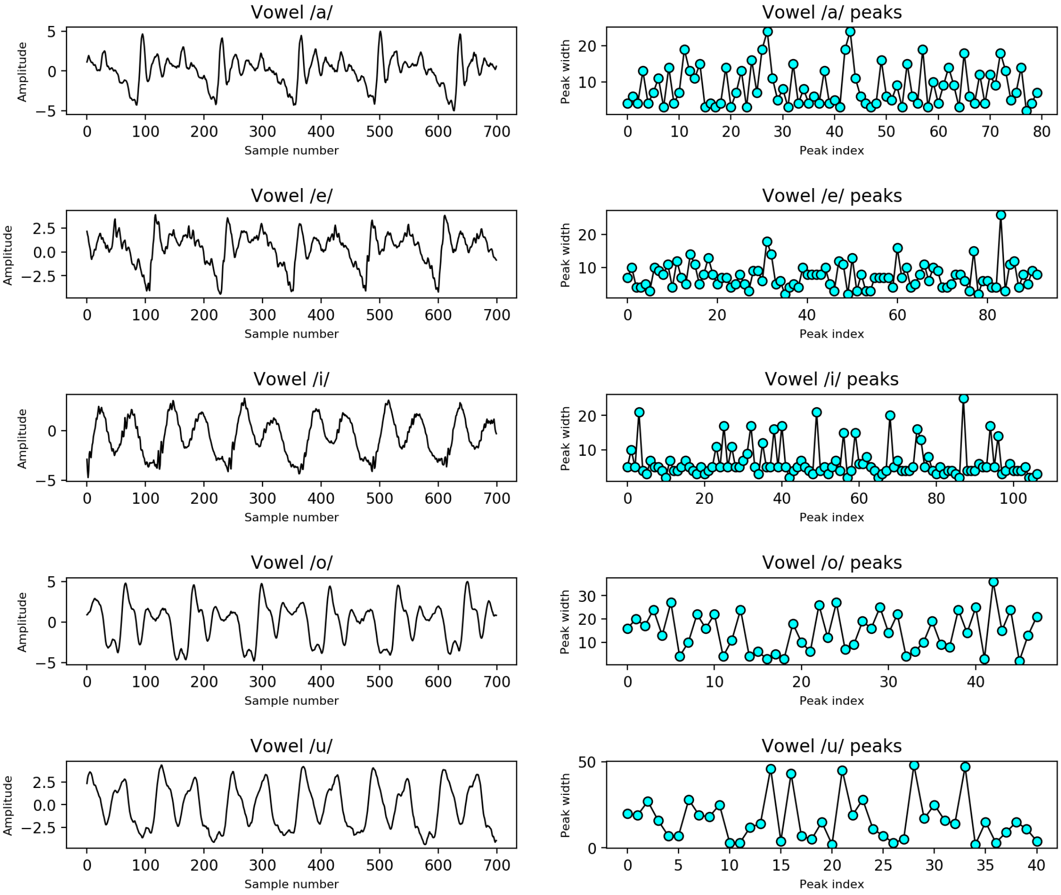


Figure 2.2: Waveforms and Peak widths of vowels /a/, /e/, /i/, /o/ and /u/ respectively

any two objects with independent motions f and g respectively. The problem is to find the smallest distance between these two objects while they move forward monotonically while preserving its orientation. This distance can be defined as Equation 2.8:

$$\delta(\tau_1, \tau_2) = Max_{f,g}|\tau_1, \tau_2| \tag{2.8}$$

Fréchet distance was originally defined for walking dog problem [26]. In the problem, a man walks with a dog where both follow different paths in the same direction but with different velocities. The constraint for movements is limited for two cases only. They can move forward or stop at any point of time as moving backward is not allowed. Therefore, the Fréchet distance between these two objects is the shortest possible length of the leash that is required to finish the walk. There are many algorithms available to solve this problem. In the present work, we used the approach proposed by Thomas and Heikki [27]. The algorithm considers three possible conditions at which man or dog can be. They are as follows:

- $Location_{man} = Location_{dog}$

- $Location_{man} < Location_{dog}$
- $Location_{man} > Location_{dog}$

In a trajectory space τ, let T_1, T_2 be two different trajectories and assume two points p_i and q_i on T_1 and T_2 respectively. Then the distance between these two points is given by Equation 2.9:

$$\delta(p_i, q_i) = max(c[p_i, q_i], min(p_{i-1}, q_{i-1})) \tag{2.9}$$

where $c[p_i, q_i]$ is the cost between objects at the present location and $min(p_{i-1}, q_{i-1})$ is the minimum cost required in the previous move. This cost covers the effort needed to travel between the points in 3 possible ways. The next point to be understood is the representation of Fréchet metric for a speech signal. For a speech trajectory τ with a sequence of acoustic events $t_i, \forall i \in \mathbb{Z}$, i.e. $\tau_s = \{T_1, T_2, ..., T_n\}$, the pattern for τ_s is defined as a sequence of similarity distance between a pair of trajectories (T_i, T_{i+1}). It is given in Equation 2.10:

$$\tau_p = \delta(T_1, T_2), \delta(T_2, T_3), ..., \delta(T_{n-1}, T_n) \tag{2.10}$$

where as $\delta : \tau_s \longrightarrow \tau_p$ is a mapping function between τ_s and τ_p. In each step, δ gives the similarity between the consecutive pair of trajectories. Thus the overall pattern of a trajectory is represented as a sequence of distances ($\delta_i s$). These values record the changes between acoustic events and thus the structural changes can be found. The procedure for feature extraction in this approach is described in Algorithm 2. The extracted similarity distance features using Algorithm 2 are used for classifying speech signals. An example of Fréchet distance is shown in Figure 2.3. From Figure 2.3-b, it can be understood that the shape of the feature vector represents the shape of source signal as shown in Figure

Algorithm 2: Fréchet distance based feature extraction

```
Input:
$S_n$: Input speech signal
$F_N$: Length of frame in samples
Output:
$FD_n$: Vector of Fréchet distances between adjacent frames
begin
    Normalize the input signal $S_n$
    Divide $S_n$ into number of frames with equal frame size
    $n_{frames} = \frac{length(S_n)}{F_N}$
    for $i \leftarrow 0$ to $n_{frames}$ do
        $FD_i \bigcup$ Fréchet distance$(T_i, T_{i+1})$    ▷ Find the Fréchet distance between each adjacent frames in the signal
    return $FD$
```

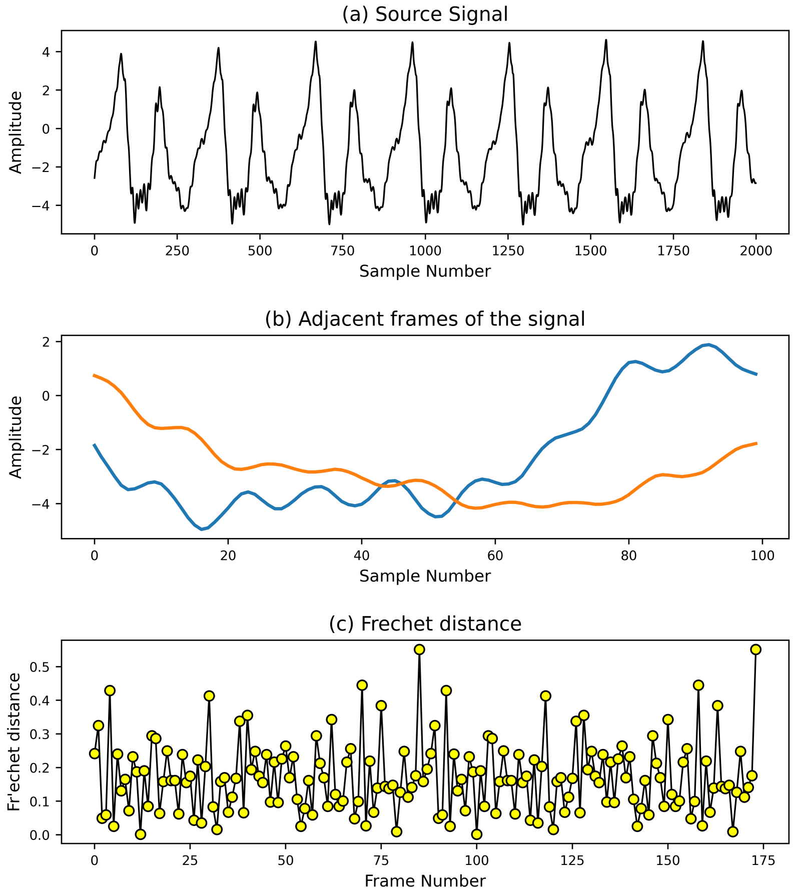


Figure 2.3: Fréchet distance calculation for vowel /u/

2.3-a. So far, we discussed the procedure for feature extraction. Next, the method that is used to model these features are discussed in the next subsection.

2.5 Speech modeling using G-HMM

Hidden Markov Model (HMM) is a statistical process in which events are hidden and observations are known. An HMM can be defined as a system with number of states and observation symbols with a set of probability functions as follows:

- State-transition probability distribution that gives probability of model being in one state and going to another state in a single step
- Observation symbol probability distribution defines the distribution of symbols for states in the system
- Initial state probability distribution defines the probability of each state being a first state in the system

A HMM assumes that the probability of a particular state depends on its previous state (Markov assumption). The probability of output observation depends only on the state from which the observation came not on any other states or observations (Conditional Independence). In G-HMM, the observation symbols follow the normal distribution. While designing a recognition system with HMM, we need to address three problems. In the first problem, the probability of observation sequence is found given the model. Second problem tries to find the state sequence from which the given observation sequence came. The third problem adjusts the model parameters so that the probability of the observation sequence is maximized. Algorithm 3 describes the sequence of steps in which a model is generated using G-HMM and the crucial parameters required in the training process are shown in Table 2.1. The HMM that was used consists of 5 states with a diagonal co-variance matrix. The total number of iterations required for the convergence is ten. Finally, Viterbi algorithm has been used for decoding the state sequence. The complete procedure for HMM is available in [28].

Algorithm 3: Model generation procedure

```
Input:
$S_n[N]$: Input speech signals
Output:
$\mu_n$: Model parameters for given speech signals
begin
    for $i \leftarrow 0$ to $N_{signals}$ do
        $\chi_i \leftarrow Extract\_Features(S_n[i])$
    $\pi \leftarrow start\_probability$
    $A \leftarrow initial\_transition\_probability$
    $\theta \leftarrow Emission\_probability$
    Initialize the means and covariance matrices
    Predict the model parameters using HMM process
```

2.6 Experimental setup

The experiments were conducted on different datasets: vowels and digits. Each data set contained 50 speaker's data. Each vowel and digit were recorded 15 times for all speakers. We have chosen speakers belonging to different regions in India. They included male and female speakers. The data was recorded using the Cool Edit software with 16KHz sampling rate, 16 bits resolution and mono channel. The data used in experiments were normalized and DC component was removed. The programs needed for experiments were implemented in Python 3.4. The libraries used are Numpy and Similarity measures [29]. The next section discusses the results observed in the study.

Table 2.1: Parameters of GHMM

Parameters	Description	Value
nComponents	Number of states in HMM	5
Covariance type	Type of the covariance matrix	Diagonal
niters	Maximum number of iterations	10
Decoder	Algorithm used	Viterbi

2.7 Results and Analysis

The study was conducted using two different features for the two different data sets as mentioned in the previous section. Each analysis is presented in subsequent subsections.

2.7.1 Peak width analysis

The aim of the analysis is to find pattern by using peak widths. An interesting characteristics observed is that vowels /a/, /e/ and /i/ have peak widths up to 20 whereas vowels /o/ and /u/ has wider peak widths. This implies the number of peak components available in vowels /a/, /e/ and /i/ are comparatively more than vowels /o/ and /u/. It also means that the temporal variations are rapid in the vowels /a/, /e/ and /i/ and it is less in /o/ and /u/. The variations observed in the feature vectors reflect the changes in the source signals. So, it is inferred that the features are significant in identifying the patterns of speech trajectories.

Peak widths are effective in steady state segments like vowels. Vowels have similar behavior over time and therefore the patterns of vowels were captured by peak width properties efficiently. Table 2.2 presents the results for intra-speaker variability and inter speaker variability. It shows that the features are useful in distinguishing vowels and digits in the intra-speaker data clearly. It gives a good classification for vowels also in the intra-speaker case.

Table 2.2: Accuracy with Peak widths

Data base	Intra Speaker	Inter speaker
Vowels	**96%**	**75%**
Digits	**89%**	58%

2.7.2 Fréchet distance-based analysis

In the study, it is found that the features are reasonable enough to characterize temporal dynamics across phonemes. As discussed in Section 2, each digit data is trained using HMM process where HMM creates models for different digits separately. The classification accuracy was tested by checking different utterances with the created models. The proposed features are effective in distinguishing the digit utterances within the speaker. That means it has the potential to classify different words. Table 2.3 gives the classification accuracy obtained in intra-speaker data for different digits. As shown in the table, experiments were conducted with varying the frame sizes starting from 80 samples to 320 samples. It can be observed that accuracy drops down after frame size of 220 samples. Frèchet distance for words "Zero" to "Nine" are shown in Figure 2.4. These graphs give an impression of structural variation among the digits clearly.

Table 2.3: Accuracy with Fréchet distance

S. No	Frame size (in samples)	Accuracy (%)
1	80	75
2	120	85
3	160	80
4	**200**	**90**
5	**220**	**90**
6	240	85
7	280	80
8	320	55

2.8 Conclusions

The chapter focuses on developing dynamic structural properties of speech signals using two different features. Peak attributes and Fréchet distance are the two features used to analyse the speech signals. In this study, it is inferred that the proposed features are useful in capturing the structural properties of spoken units which is useful for classification. One of the advantages of peak attributes is the extraction procedure which is simple to compute. When compared with standard features LPCC and MFCC where a frame of the speech signal is transformed to a vector, this method gives a simple representation. Here, the entire signal is considered and it is transformed into a vector that contains peak attributes. Another advantage is that the continuous temporal pattern can be extracted in one pass without losing intra-segmental clues. Even though the method is not highly accurate, it gives good accuracy for vowel classification and digit classification comparatively with the existing parametric trajectory segmental models (approximately 75%) [30] [3]. The modeling technique that was employed can be improved by considering the individual model characteristics of different speakers. This will be the future scope of the present work.

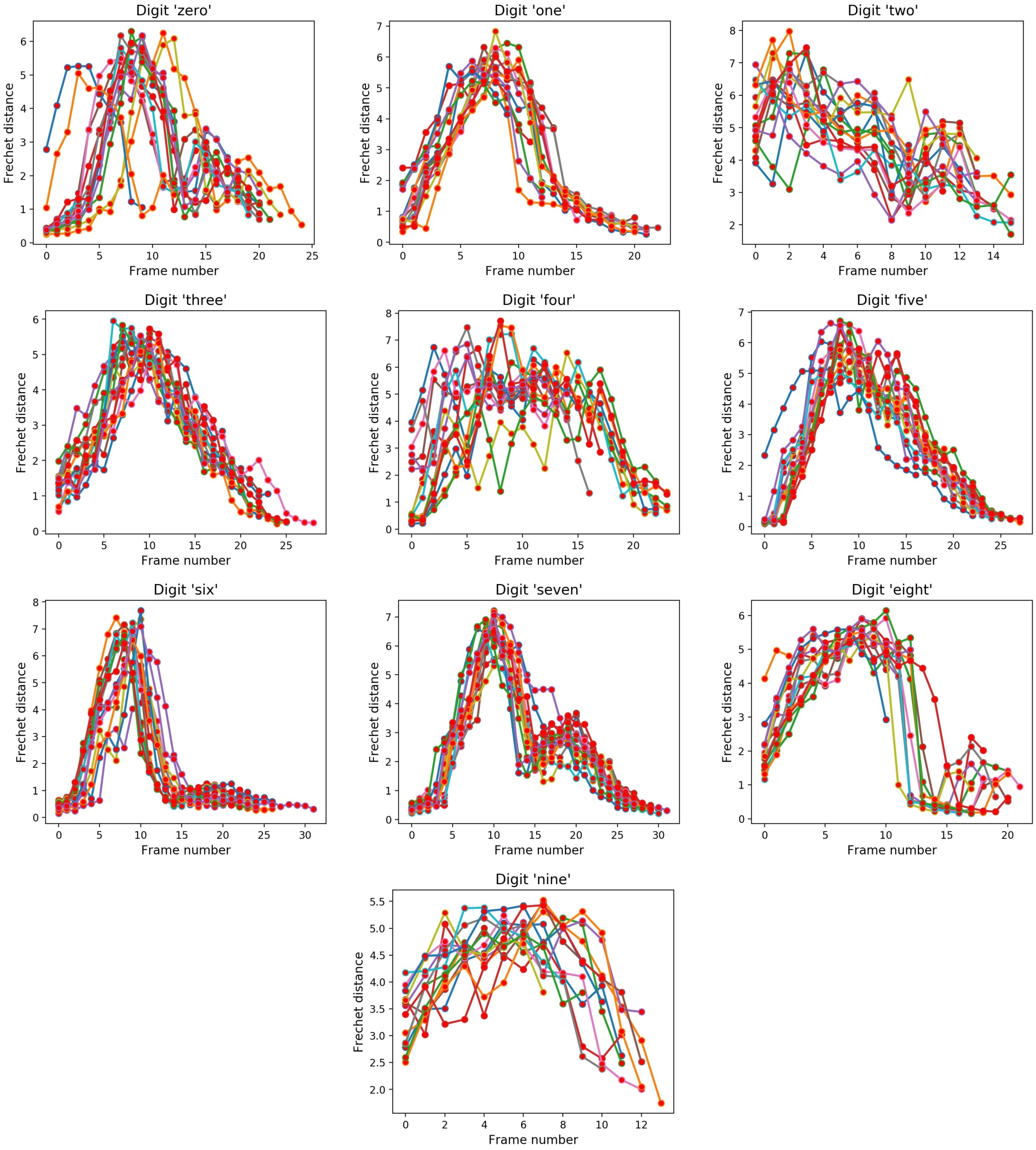


Figure 2.4: Fréchet distance of the words "Zero" to "Nine"

Chapter 3

Characterization of Spoken English vowels using Tree structures

The previous chapter discussed the features that are based on trajectory attributes. The features were trained using G-HMM and the approach has been shown to be useful for classifying vowels and digits. HMMs are efficient in modeling speech signals, but they require complex computations and huge data for training. But for small set of speech data, we cannot go for complicated tasks. Therefore, a simple process that can give an efficient representation for spoken units is useful. In this direction, we propose a new data representation for speech signals. The motivation for the work is the latest developments in signal processing domain called Graph Signal Processing (GSP). Recent trends have shown that the research is focusing on combining the approaches in signal processing and graph theory [31]. This new area of interest addresses the study of irregular structures found in different domains like social networks, citation networks, etc. There are different signal processing concepts that have been used in this problem domain. Here we address the problem of speech signal analysis using graph structures. The representation was defined for spoken vowels of Indian accented English. The work in this chapter has been published in [32]. A brief explanation for vowel characterization is given in Section 3.1 and a related work on tree structures is given in Section 3.2. The complete procedure is discussed in Section 3.3. The Experimental details are given in Sections 3.4 and 3.5.

3.1 Motivation

Characterization of vowels in spoken English sentences plays a significant role in designing speech processing systems. In this work, spoken English vowels are analysed to find features which can help in their characterization. The outcome of the analysis led to the proposal of a novel feature representation called tree structures for vowels. In this approach, the vowels are represented as trees with their structural properties being elements in the trees. These properties are extracted by understanding the geometrical shapes of acoustic events found in waveforms. To prove the effectiveness of features, a tree comparison is shown by calculating the tree distances. The computation of distance is done by employing a tree matching algorithm. The performance of the proposed features are compared against the standard MFCC features. In the analysis, speech data of Indian native speakers was used. The analysis procedures and the results obtained are presented. In the area of speech recognition, the content in speech signal is understood to extract meaningful features that can help subsequently in the recognition task. To accomplish this, spectral feature extraction techniques like LPCC [33][34], MFCC [35], fundamental frequency, formants, etc. and temporal features like energy, ZCR (Zero Crossing Rate), pitch, etc. are found to be popularly used features in the speech recognition domain. In the next subsection, a brief overview of tree structures is given.

3.2 Tree structures

The concept of waveform processing using tree structures was first proposed by Ehrich [36]. The goal of tree structure is to construct a representation that reflects the spatial structure of a waveform. The assumption here is that the necessary information can be found in the peaks and valleys of the obtained waveform of a speech signal. Tree structures are constructed by scanning a waveform segment from left to right. The obtained peaks and valleys are inserted into a tree structure in such a way that interpretation of the tree can give meaningful information of the underlying waveform. This form of structure is called a Relational tree since it not only contains the basic elements of the waveform, but also the relationship among them. Each tree representation gives a unique pattern which is the sequence of peaks in some order. This structure was studied and further modified by Lu and Cheng by adding amplitude information of peaks. The resultant structure was known as Skeletal tree [37]. Subsequently, it was modified to include temporal information. The tree with all these quantities was called a Complete tree. The concept of structural representation to process waveforms was used by Shaw et al. [38] to recognize structural similarity in seismic and ECG classification. Waveforms are classified into different types of signals by using a distance measurement. The tree structures have been used by different researchers in domains like image profiling, handwritten signature verification [39], text to speech synthesis(TTS) [40] and prosody modeling [41]. Fisher and Ritchings proposed Attributed Relational tree [42] in which each node i.e. peak is associated with amplitude and width. This method was used to characterize features that are extracted from the waveform image profile. In the present work, the relational tree approach is examined to observe the nature of vowels sounds and subsequently to characterize them. The detailed method used in this work is presented in the next section.

3.3 Proposed method for vowel characterization

The whole framework consists of 4 major steps. They are listed as follows:

1. Pre-processing
2. Segmentation and Averaging
3. Tree structure generation
4. Tree comparison

In the first step, pre-processing the input signal is normalized to suppress the effect of DC (Direct Current) component. Next, the normalized signal is segmented into a number of frames with fixed length. The size of the frame is decided based on the approximate pitch period of the input signals. Then the frames are processed to get the average signal. These average signals are considered for constructing the tree structure. There are two different types of trees for representing peaks and valleys separately. The procedures used for tree construction are discussed in subsections 3.3.1 and 3.3.2. Finally, the generated trees are compared to classify the vowels. The procedure for tree comparison is discussed in Section 3.3.3. The primitives that are defined in Section 2.4.1 are used in this procedure.

3.3.1 Construction of Peak Tree

Next we define tree structures used in the present approach. Two structures have been used as mentioned earlier for representing peaks and valleys individually. Peak tree contains the peaks as basic nodes whereas valley tree contains valleys of the waveform. In each peak and valley tree, the edges represent the relationship between peaks and valleys respectively. The Peak tree contains the structure of the waveform in terms of changes pertaining to the peaks. The procedure used for constructing this tree is described as follows:

1. For each signal segment:
   - Locate all the peaks in the frame
   - Mark the highest peak and create a node in the tree. This node is the parent node of a tree. The node divides tree structure into two parts, one in left side of the parent node and another in right side of the parent. Highest peak is computed using Algorithm 4 using the function *Max()*. *Createnode()* method inserts a node into the peak tree.
   - The left subtree and right subtree incorporate peaks information of left and right parts respectively.
2. Repeat Step-1 for left partition and right partition until all the peaks are inserted as nodes in the tree.

```
Algorithm 4: Peak tree construction
Input:
root: root node of the peak tree
peakarray: array of peak points
lb: lower bound of the peakarray
ub: upper bound of the peakarray
Output:
root: Root of the created peak tree
begin
    CreatePeakTree $(root, peakarray, lb, ub)$
    if $lb < ub$ then
        $mid \leftarrow$ Max$(peakarray)$
        $root \leftarrow$ Createnode$(mid)$
        $root.left \leftarrow$ CreatePeakTree$(root.left, peakarray, lb, mid - 1)$
        $root.right \leftarrow$ CreatePeakTree($root.right, peakarray, mid + 1, ub$)
        return $root$
```

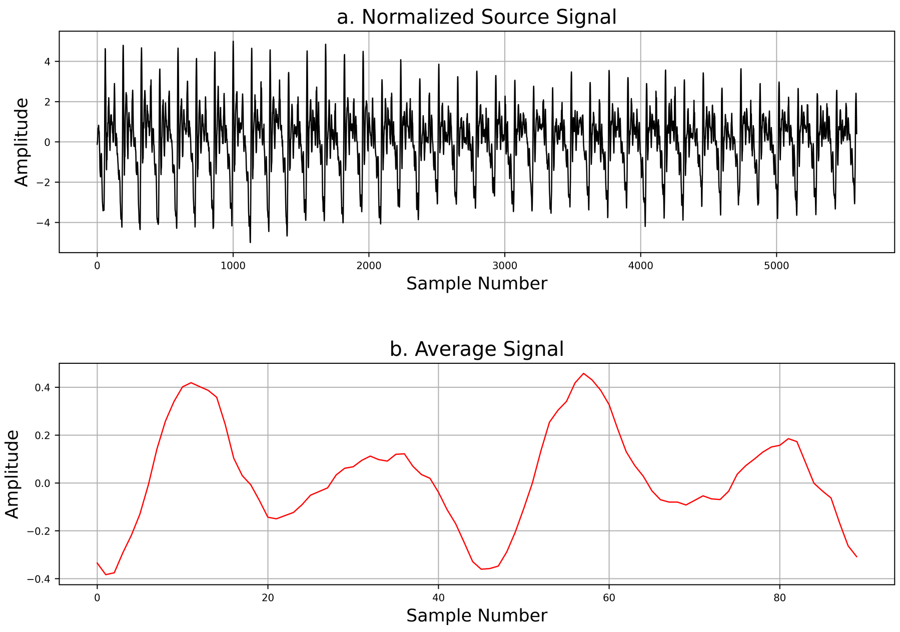
The algorithm works in a recursive fashion. To understand the process of peak construction, consider an example. The normalized speech signal of Vowel /a/ is shown in Figure 3.1-a . As discussed in procedure, this signal is segmented into frames to get the average signal. Then, the average signal is processed to extract peaks and valleys from which trees will be generated subsequently. The average signal for the vowel /a/ is given in Figure 3.1-b and the trees generated for the same are shown in Figure 3.2. Figure 3.3 shows average signals for each vowel along with their source signals. Algorithm 4 gives the detailed steps of the peak tree construction. The procedure for the valley tree construction is explained in the next subsection.

3.3.2 Construction of Valley Tree

Valley tree represents the properties of valleys in a waveform. The procedure for constructing the tree is similar to peak tree generation. Here, the deepest valley in the signal becomes the key element for partitioning. The recursive procedure for valley tree creation is presented in Algorithm 5. In this algorithm, *Min()* returns the deepest valley among the valleys.

3.3.3 Tree Comparison

The final step in the approach is comparing the trees. In this phase, the tree structures obtained for each vowel by the procedure discussed in Sections 3.3.1 and 3.3.2 are used for comparison. For the purpose of comparison, a tree edit distance algorithm has been used. It finds the similarity between any two trees by calculating their edit distance.


a. Normalized Source Signal

b. Average Signal

Figure 3.1: Normalized signal and its corresponding average signal

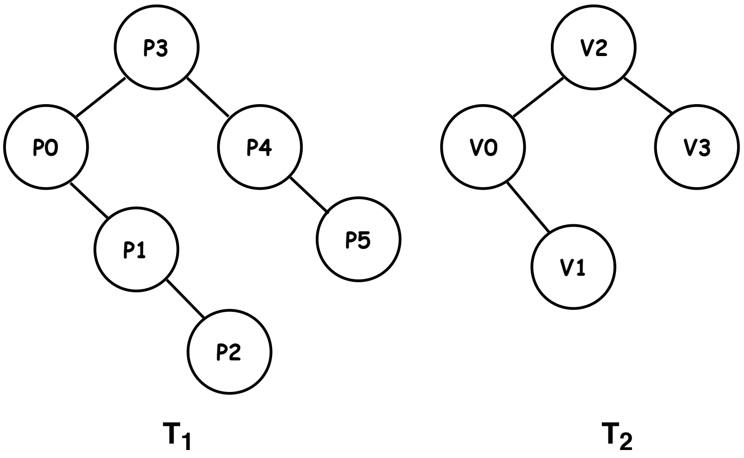


Figure 3.2: Tree structures for Fig. 2 (T1 - Peak tree, T2 - Valley tree)

The edit distance between a pair of trees T_1 and T_2 is the minimal cost required for edit operations to transform T_1 to T_2. The elementary operations required for this task are as follows:

- Substitution - replaces label of a node
- Insertion - inserts a new node into tree
- Deletion - deletes an existing node from tree

Let (M, T_1, T_2) be a mapping, the cost of M can be given by Equation 3.1. Edit distance

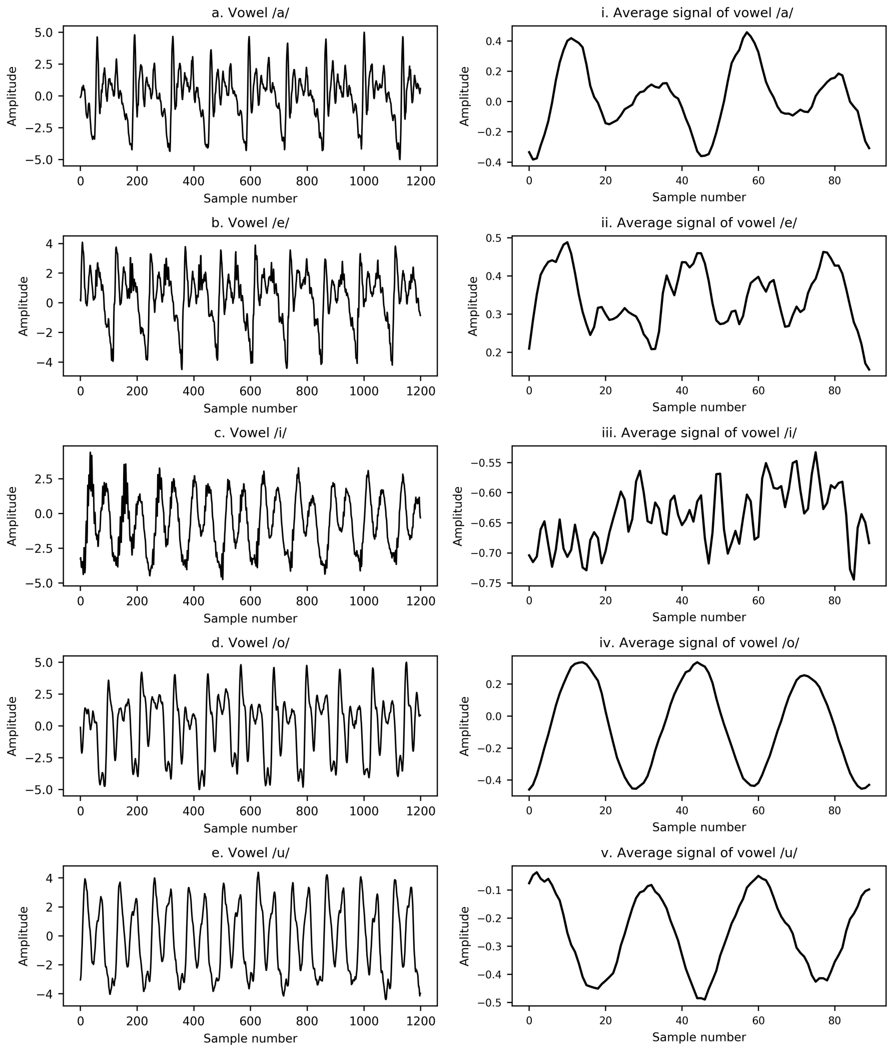


Figure 3.3: Source signal and average signals of vowels

can be computed using Equation 3.2 [43].

$$\begin{aligned}\gamma(M) = \sum_{(v,w)\epsilon M} \gamma(v \to w) + \sum_{v\epsilon N_1} \gamma(v \to \lambda) \\ + \sum_{w\epsilon N_2} \gamma(\lambda \to w)\end{aligned} \tag{3.1}$$

$$D(T_1, T_2) = min\{\gamma(M)|(M, T_1, T_2)\} \tag{3.2}$$

where:

- N_1 is the set of nodes in T_1
- N_2 is the set of nodes in T_2

Algorithm 5: Valley tree construction

```
Input:
root: root node of the valley tree
valleyarray: array of valley points
lb: lower bound of the valleyarray
ub: upper bound of the valleyarray
Output:
root: Root of the created valley tree
begin
    CreateValleyTree $(root, valleyarray, lb, ub)$
    if $lb < ub$ then
        $mid \leftarrow$ Min$(valleyarray)$
        $root \leftarrow$ Createnode$(mid)$
        $root.left \leftarrow$ CreateValleyTree$(root.left, valleyarray, lb, mid - 1)$
        $root.right \leftarrow$ CreateValleyTree($root.right, valleyarray, mid + 1, ub$)
        return root
```

- $v \to w$, $v \to \lambda$, $\lambda \to w$ are edit operations

The edit distance algorithm that has been used in the present approach was proposed by Zhang and Shasha. Apart from this, there are other methods available in literature [44]. The motivation to use Zhang edit distance is its computation efficiency and its ability to compare ordered trees. There are two significant properties for any ordered tree matching algorithm. They are as follows:

1. Relation between root and child
2. Sibling order

We are not discussing the detailed steps of this algorithm here. The complete steps in distance matching algorithm can be found in [45]. So far, we have seen the crucial steps in the present approach. In the next section, the environment in which the experiments have been conducted is discussed.

3.4 Experimental setup

In this section, we present the environment used for conducting our experiments. The algorithms have been implemented using the Java programming language. We used 20 speaker's data for the analysis. Each English vowel is recorded 10 times for all speakers. We have chosen speakers belonging to different regions in India. They include male and female speakers. The vowels were recorded using the Cool Edit software with 16KHz sampling rate and mono channel and 16bits/sample. The data used in experiments were normalized and DC component was removed. Each waveform is divided into number of frames of fixed length. We used window size of 95 samples for the experiments. The results are discussed in the next section.

3.5 Results and Discussion

To summarize, the study has been focused on extracting phonological structures from the waveform through tree structures. For showing the evidence that the proposed structures are efficient to represent vowel sounds, intra speaker variability and inter speaker variability are examined. This section is divided into two subsections. In Subsection 3.5.1, the analysis concentrates on how well the variability among the vowels holds good. And in Subsection 3.5.2, the analysis of vowels features over different speakers are shown. Finally, the nature of patterns in noise conditions are also presented in Subsection 3.5.3.

Table 3.1: Edit Distance between tree structures of each vowel for Speaker 10

	Peak Tree Distances					**Valley Tree Distances**				
Vowels	**/a/**	**/e/**	**/i/**	**/o/**	**/u/**	**/a/**	**/e/**	**/i/**	**/o/**	**/u/**
/a/	**9**	17	18	16	11	**5**	12	9	7	9
/e/	15	**10**	21	15	15	12	**9**	8	8	15
/i/	20	17	**15**	20	18	9	11	**6**	8	5
/o/	9	17	19	**4**	12	8	9	8	**4**	6
/u/	11	17	20	11	**13**	7	10	7	3	**6**

3.5.1 Intra Speaker Analysis of tree structures

The significance and interpretation of tree structures depend on two factors as follows:

1. Meaning of individual peak and valley trees in pattern analysis
2. Suitability of tree structures for spoken vowels

The former issue can be addressed by carefully understanding the structure itself. Each peak (or valley) tree contains the order of the peaks (or valleys) in terms of their occurrence in a waveform. This encompasses the priorities of each peak (or valley) in a speech signal from the highest to lowest. Therefore, the highest peak comes as root node in peak tree and deepest valley appears as root node in valley tree.

When a tree structure is visited, each traversal (pre-order, post-order, in-order) of tree gives different patterns of the underlying trees. These patterns correlate the changes in the waveform at different instances. Suppose, a peak P_i with height X appears as root node of a tree T_1, that means P_1 is the highest. There are various ways of arranging the same P_i in the tree. These different arrangements give a way to represent different orders in which the components (peaks, valleys) can be placed. So the changes in waveform of different vowels can be easily represented in the proposed structure. Now to answer the suitability of the present approach, the outcome of the study has to be considered. There are enough clues found by the experiments that these spoken units can be used for vowel classification.

Intra vowel analysis is used to understand the similarities within the vowels. It gives an insight into the nature of tree structures for different trees of the vowels. In the comparison, both the peak trees and valley trees have been used. The edit distances between tree structures of each vowel are shown in Table 3.2. These distances are the average edit distances among the 5 utterances of each vowel for a speaker. The important observations are as follows:

1. From Table 3.1, we can understand that the edit distances between peak trees of respective vowels is the lowest among others except in case of vowel /i/. It has collision properties of vowels /e/ and /o/. That means, the similarity between the same vowels hold.
2. In case of valley tree representation, similar observations can be seen. Vowel /a/ has shared similarity with vowel /u/ and the vowel /i/ has similarity with vowel /u/.
3. The distances between the valley trees are comparatively less while distance in peak trees is more. Thus valley trees of vowels belonging to same speaker are more similar than peak trees.

From the above mentioned inferences, it can be understood that valley tree representations are better than peak trees for distinguishing within the same vowels.

3.5.2 Inter Vowel Analysis of tree structures

The tree edit distances between each vowel belonging to same speaker are compared to find the suitability of tree structural representations for vowels. For this, both the tree structures have been used. The selection of window size for the analysis is made on empirical observations. It is observed that tree structural differences among the vowels are distinguishable well in case of frame size 95 over the frame-sizes 75, 80, 85, 90, 95, 100, 120 and 135. The results of this analysis is presented for each vowel separately. It is found that vowel /a/ has peak trees which has separate structures compared to another vowels in 67% of speakers. There are collisions with other vowels /o/ and /u/ in remaining cases. But the distance is found to be less. The variations in the valley trees of vowel /a/ are not clear enough to discriminate from other vowel structures. But the cases in which peak tree has failed, valley tree is able to give good discrimination. Therefore, the analysis has shown that better differentiation can be made by combining both the peak trees

Table 3.2: Edit Distances between tree structures of each vowel for different speakers

	Peak Tree Distances					Valley Tree Distances				
S.No	/a/	/e/	/i/	/o/	/u/	/a/	/e/	/i/	/o/	/u/
1	3	8	12	2	3	2	6	8	1	3
2	4	11	16	3	9	4	8	6	2	3
3	4	2	4	4	5	3	3	4	4	4
4	5	13	15	2	3	3	9	10	1	3
5	10	12	12	6	2	5	10	10	4	1
6	9	12	12	6	4	6	10	7	5	1
7	5	7	10	3	4	3	7	9	2	3
8	7	9	10	3	2	6	8	9	1	1
9	5	9	9	5	4	4	8	9	4	3
10	5	13	13	6	4	3	10	7	4	1
11	6	10	7	6	4	5	7	6	4	1
12	8	13	16	4	5	6	8	8	3	3

and valley trees. The combined approach is able to recognize 70% of speakers. In case of vowel /e/, 42% speakers have peak trees and valley trees that are different to other vowels. In this case also the valley trees and peak trees can be combined to differentiate the vowel structures. The combined approach gives 50% accuracy. The detailed average edit distances between vowels of different speakers are given in Table 3.2. The accuracy for each vowel in 3 different cases i.e. valley tree, peak tree and combined approach is shown in Table 3.3. It is found that similarity in tree structures of valley is high compared to the similarity in tree structures of peaks for different vowels. Even though distinction can be drawn by the peak trees alone, valley trees play a key role to distinguish among the vowels as it shows less similarity in some speakers where peak trees are unable to discriminate. With peak tree, we can make clear distinction between vowels. But comparison of vowels /o/ and /u/ shows that there is similarity in patterns of tree structures between these two vowel sounds.

Table 3.3: Inter vowel analysis

Vowel	Peak trees	Valley trees	Combined approach
/a/	67%	25%	**70%**
/e/	42%	42%	**50%**
/i/	50%	42%	**67%**
/o/	58%	58%	**66%**
/u/	75%	67%	**75%**

3.5.3 Nature of features in the presence of noise

In this section, the performance of the proposed approach for noisy signals is discussed. The experiments have been conducted on noisy signals with Signal-to-Noise Ratio (SNR) of 20dB. The type of noise is random white noise that has a normal or continuous distribution [46]. The results found are interesting. The nature of these tree structures could show stability over noisy signals also. The reason behind this is the special property of monotonic scaling along the time [38]. It makes the structure stable even if there are some perturbations occurring in the input signal. The comparison table for speaker 10 in noisy signal environment is shown in Table 3.4. For the same speaker, the results have been shown in Table 3.1 for clean speech signals. It can be seen that the essential features that distinguish the vowels still hold in the tree structures.

Table 3.4: Edit Distance between tree structures of each vowel for Speaker 10

	Peak Tree Distances					**Valley Tree Distances**				
Vowels	**/a/**	**/e/**	**/i/**	**/o/**	**/u/**	**/a/**	**/e/**	**/i/**	**/o/**	**/u/**
/a/	**9**	17	18	16	11	**5**	12	9	7	9
/e/	15	**10**	21	15	15	12	**9**	8	8	15
/i/	20	17	**15**	20	18	9	11	**6**	8	5
/o/	9	17	19	**4**	12	8	9	8	**4**	6
/u/	11	17	20	11	**13**	7	10	7	3	**6**

3.6 Conclusions

In the present work, the vowel characterization problem has been addressed by a structural processing technique. The idea is to treat a speech signal as an image instead of a time series. To represent the temporal structure of a speech signal, tree structures have been used. The detailed procedures for the tree construction and analysis were presented. The results obtained in this process is promising even though it is not competing with the current methods. The classification rate of parametric trajectory segmental models has been 72% approximately [47] in case of vowels. This system used Mel Frequency Cepstral Coefficients (MFCC) as features and segmental HMMs. Li. Deng reported that the system with Hidden Dynamic Models (HDMs) with frequency warped LPCCs give accuracy of 75% [48]. When the proposed approach is compared with the above mentioned approaches, it does not give superior performance than them. Still there are possibilities to improve the method by considering other properties of the speech signal. For example, we considered only the order of peaks and valleys. But additional patterns or clues can be found by concentrating on positions, width and distance between adjacent peaks (or valleys). To make the methods more useful in recognition tasks, models can be generated with available graph learning algorithms.

This chapter discusses a method that understands the structure of a phonetic unit towards classification. The properties of signal's shape is not only pivotal in its characterization, but also for phoneme boundary analysis. A study on this problem is discussed in Chapter 4.

Chapter 4

SPEECH TRAJECTORIES FOR PHONEME SEGMENTATION

In the previous chapters, the geometrical methods for signal characterization towards recognition were discussed. In these methods, the attributes of the signal's shape were utilized. Another important problem in speech processing domain is speech segmentation or phoneme boundary detection. This task helps in improving the recognition quality by providing proper segmentation information for phonemes or phonetic units. It is an important step as inappropriate segmentation may lead to recognition accuracy falloff. The problem is essential not only for recognition but also for annotation purposes also. In general, segmentation algorithms rely on large data sets for training where data is observed to find the patterns among them. But this process is not straight forward for languages that are under resourced because of less availability of data sets. In this chapter, a method that uses geometrical properties of waveform trajectory where intra-signal variations are studied and are used for segmentation. The geometric properties are extracted as linear structural changes in a raw waveform. The method works by extracting useful attributes of the signal's shape and these properties are combined further to find the segmentation points. A correlation algorithm called Canonical Correlation Analysis (CCA) is used to study the combined geometrical features. The data used in the analysis are Indian accented English words. Finally it is found that the proposed approach is useful in the segmentation task. This chapter is organized as follows: The next section describes trajectory methods that were used for pattern analysis. Section 4.2 gives an overview of the CCA method. Section 4.3 explains the proposed approach for segmentation. The data and experimental setup is described in Section 4.4. Section 4.5 explains the results found in the study and Section 4.6 concludes the chapter.

4.1 Trajectories for pattern analysis

In an Euclidean space, a trajectory is defined as a curve that is formed by the observation of the path that a moving object makes. The points in the path are characterized as ordered positional points. Trajectory models that were initially known as Linear Trajectory Segmental Models (LTSMs) have been used to analyze speech signals for past 3 decades [13]. Trajectories are suitable in pattern analysis for two reasons [49]:

1. A speech trajectory is influenced by the context of the spoken words
2. Trajectories formed by different phonetic units can form independent clusters based on the contextual information

However the models that are based on HMM are suitable for large vocabulary speech recognition [50]. Trajectories are not only used for speech signal analysis, but also for pattern analysis in different areas like road network [51], databases [52], traffic management, etc.

A trajectory contains vital information like spatiality and temporal patterns about an object. There can be different ways of treating trajectories: segments sequence and points sequence. The similarity metrics to measure the affinity vary on the kind of trajectory. The effectiveness of the comparison method depends on the underlying components that the trajectory represent. Huanhuan et al. proposed a fusion based similarity method for traffic flow patterns [24]. The method combines different techniques like Merge Distance (MD), Multi Dimensional Scaling (MDS) and Density based Spatial Clustering of applications with noise (DBSCAN) to identify traffic flow patterns and customary routes from vehicle movements. One of the fusion techniques is given by Equation 2.5. Next section describes the CCA technique.

4.2 Canonical Correlation Analysis (CCA)

Canonical correlation analysis (CCA) was introduced by Hoteling for multi-variate analysis. It helps to find the relation between multiple variables simultaneously that makes analysis easy. The critical step in CCA is to find a set of transforming variables that can transform variables such that the transformation in the corresponding new coordinates is maximally correlated. In the process, a set of variables called as canonical weights are used. The solution to this is computationally expensive and time consuming. Therefore, it is convenient to solve the problem as an eigen value problem. The objective function to solve CCA for two variables x and y can be expressed by Equation 4.1:

$$C = \begin{pmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \rho^2 \begin{pmatrix} C_{xx} & 0 \\ 0 & C_{xx} \end{pmatrix} \tag{4.1}$$

where C_{xy} is the covariance between variables and C_{xx}, C_{yy} are auto covariances of variables x and y respectively. There are various applications of CCA in the signal processing domain. It has been useful in finding relations which can help in multi-view learning [53]. Heycem et.al. applied the technique for feature selection for the problem of depression recognition from speech signals [54]. Wang et.al. used CCA to learn acoustic features that can improve phonetic recognition [55]. Apart from the above mentioned applications, CCA is also useful in areas like Blind Source Separation (BSS). In this problem, the aim is to recover the original signal when an unknown linear mixture of statistically independent signals are available [56]. Another approach based on CCA focuses on improving the signal to noise ratio (SNR) in EEG data signals that are recorded from multiple channels [57].

In the present work, knowledge from a set of multiple features is used to detect boundary points in a word. The complete procedure is explained in Section 4.3.

4.3 Proposed approach for segmentation

The proposed method uses cumulative knowledge of multiple geometric features and combine these to form a multi-view trajectory feature vector. The feature vector is then analyzed dynamically to extract phonetic boundaries. The crucial constituents of the approach are as follows:

1. Basic feature set (τ)
2. Derived features (τ_D)
3. Multi-view boundary detection algorithm

Each component is explained in the following subsections. Basic and derived features are defined in the next subsection. The segmentation algorithm is explained in Section 4.3.2.

4.3.1 Trajectory features

A speech signal records the nature of vibrations when the vocal chord moves for uttering a sound. The resultant waveform consists of peaks and valleys which helps to understand the salient features of the spoken sound and the speech characteristics of the person who has uttered that sound. Thus the waveform records different acoustic events which can be used for various purposes like classification, segmentation, etc. One of the crucial nature of a trajectory is its shape. Each event that is recorded in a speech signal has a distinct structure. The structural properties of phonetic units have become an interesting area of study [4]. The reason for this is that the features correspond to phonetic characteristics with variations in a lucid way. And also the structural properties of waveform trajectories are useful in understanding the dynamic nature of different phonetic units. In the present work, a set of geometric features are proposed to capture the transitional behavior of the waveform that can be further used in identifying boundary points between different phonetic units. The feature set as a whole contains two different classes i.e. primitive and derived properties. The primitive properties are those characteristics that are inherent in a waveform. They are listed as follows:

1. Peak
2. Valley
3. Peak position
4. Valley position

In the second stage, the aforementioned features are transformed further to obtain derived attributes. This set contains the following elements:

1. Peak width
2. Valley width
3. Slope of peaks and valleys
4. Disparity of peaks and valleys

For a segment of speech signal $S[n]$ with size m, the terms are defined in Definitions 4 to 9.

Definition 4 **Peak position** is any integer k, such that $0 < k < m$ where peak is found at k^{th} location

Definition 5 **Valley position** is any integer k, such that $0 < k < m$ where valley is found at k^{th} location

Definition 6 The data point p_k being a peak point between the valleys v_q and v_r, the difference $r - q$ is defined as **peak width** for the peak p_k $\forall k, q, r \in \mathbb{Z}$ and $q < k < r$

Definition 7 The data point v_k being a valley point between two peaks p_q and p_r, the difference $r - q$ is defined as **Valley width** of valley v_k $\forall k, q, r \in \mathbb{Z}$ and $q < k < r$

Definition 8 The **slope** between two points $x = (x_1, y_1)$ and $y = (x_2, y_2)$ is defined by Equation 4.2.

$$Slope(x, y) = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.2}$$

Definition 9 The **Disparity** between two points p_i and p_k is given by Equation 4.3.

$$Disparity(p_i, p_k) = \sqrt{(p_i - p_k)^2}, \forall i, k \in \mathbb{Z} \tag{4.3}$$

To understand the terms, let us consider Figure 4.1. In the figure, peaks and valleys are indicated as P_i and V_i respectively where i represents the sequence in which they occur in a waveform. The next term, peak-width is the width of the curve in a waveform between two valley positions. In the same way, valley width is the distance between two peaks in which a valley is present. Slope is the general gradient between two points in a geometric space. The points that are considered here are a pair of peaks (or valleys). This feature gives information of two adjacent peaks (or valleys). In the segmentation algorithm, the average slope between peaks (and valleys) of each frame in the source signal is studied. Finally, the property 'Disparity' between two points (peaks or valleys) is the continuous variation between the heights of peaks and depth of valleys.

The property 'slope' considers the position at which the peaks (or valleys) occur whereas 'Disparity' does not regard this property. The derived features of the word "Zero" are shown in Figure 4.2. Figure 4.2-a shows the source signal in normalized form and Figure 4.2-b, Figure 4.2-c give slope of peaks and disparity respectively. Slope and disparity of valleys are shown in Figure 4.2-d and Figure 4.2-e respectively. The procedure used for segmentation is explained in next subsection.

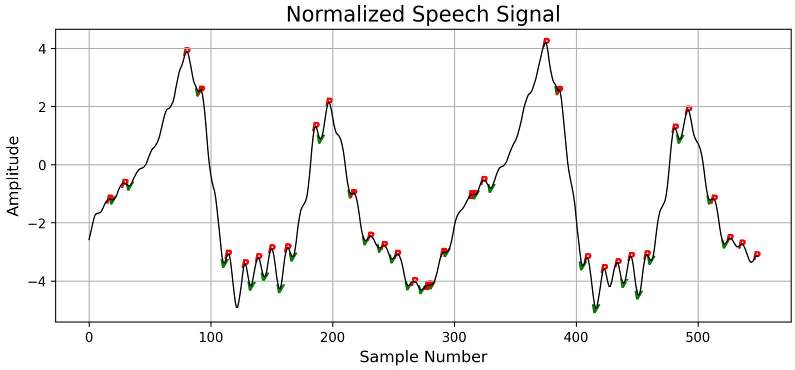


Figure 4.1: Peaks and valleys of a segment of speech segment

4.3.2 Multi-view boundary detection algorithm

The features that are described in the previous section are analyzed to understand the boundaries of the phonetic units. The algorithm observes the dynamic changes of the waveform over the entire signal by capturing the variations of the waveform with the extracted features. First, the given speech signal is divided into equal-sized frames. A set of basic features (τ) are extracted from each signal. From the basic features, a set of derived features are drawn. Thus the complete feature set is a matrix in which each set of derived features are present. This is a multi-view representation of the waveform trajectory features that will be processed to find the segmentation points. The segmentation procedure comprises of two stages: In the first stage, the feature matrix is analyzed by the CCA procedure which will give a set of coefficients for each feature set simultaneously. These coefficients represent the correlation between subsets of each feature set which will be used next. In the second stage, a pair of sequential frames that are adjacent will be used to generate correlation coefficients. Finally, the coefficients generated in first and second stages are then compared to get the variance between them. The crucial steps in the segmentation procedure can be summarized as follows:

1. The input signal $S[n]$ is divided into a set of frames $f_0, f_1, ..., f_n$ of equal size.
2. Each frame is then transformed to a set of primitive features : S_p, S_v, S_{pi}, V_{vi}, where:
   - S_p is set of peaks

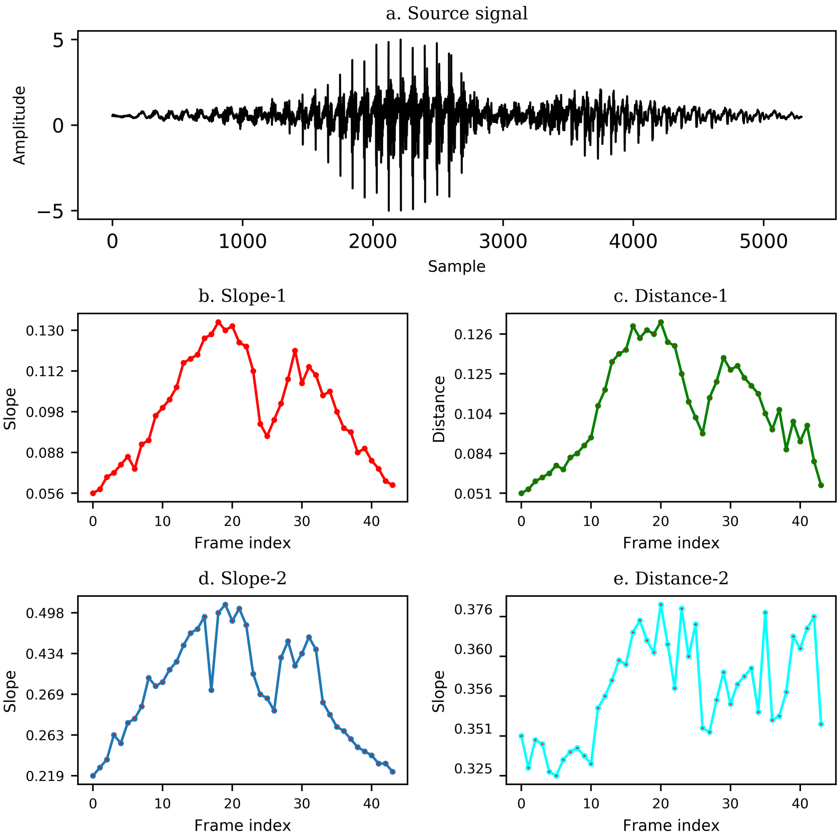


Figure 4.2: Derived Peak attributes (Slope, Disparity) for the word "zero"

- S_v is set of valleys
- S_{pi} is set of integers that represent peak positions
- V_{vi} is set of integers that represent valley positions

3. The features obtained in Step 2 are then transformed to a set of trajectory features $\tau = \tau_{sv}, \tau_{sp}, \tau_{dpv}, \tau_{dp}$.

4. The feature sets τ are analyzed using CCA which gives a set of coefficients represented by CCA_{τ}.

5. The features sets belonging to subsequent frames are correlated to get the new coefficients. Each set consists of features belonging to 3 adjacent frames. The number of frames is empirically chosen so that variations can be captured in the corresponding CCA coefficients.

6. Variance between coefficients computed in Step 4 and Step 5 are compared. The peaks in this set forms the boundary points. Thus the peaks in each set are combined to identify the boundary points using the CCA_{τ} computed by Equation 4.4.

$$B_p = \{CCA_{\tau_{dp}} \cup CCA_{\tau_{dpv}} \cup CCA_{\tau_{sp}} \cup CCA_{\tau_{sv}}\} \tag{4.4}$$

The final variances obtained for each derived feature set are shown in Figure 4.3. From

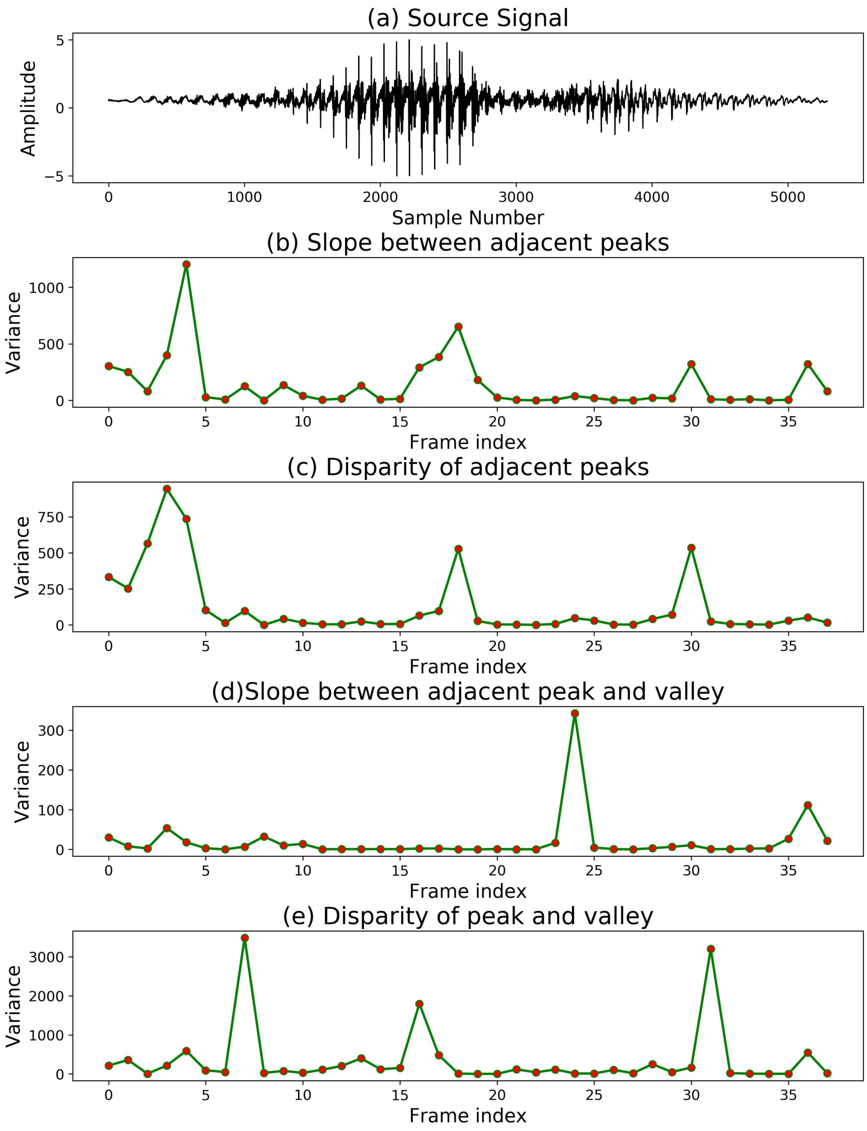


Figure 4.3: CCA variance of different features for the word "Zero"

the diagram, it can be observed that the changes needed for identifying the phonemic variations are recorded in as peak points in the final variances. But different types of variations can be seen separately from the features. Therefore it is required to combine the points obtained from each feature to get the final boundary points. The detailed algorithm is given in Algorithm 6 and the flowchart is shown in Figure A.1. In the next section, the background setup used for the experiments is described.

Algorithm 6: Boundary detection algorithm

```
Input:
$S[n]$: Speech segment of length $n$
$k$: Size of the frame
Output:
$BP$: Boundary points of phonetic units
begin
    Step 1: Normalize S[n]
    Step 2: Divide S[n] into frames with equal size $k$
    Step 3: Let $F_n$ be number of frames
    for $i \leftarrow 0$ to $F_n$ do
        Step 3.1: Find peaks using Definition 1
        Step 3.2: Find valleys using Definition 2
    Step 4: for $i \leftarrow 0$ to $F_n$ do
        for $j \leftarrow 0$ to $Max(n_{peaks}, n_{valleys})$ do
            Step 4.1
            $T_{sp} \leftarrow Slope(peaks_j, peaks_{j+1})$
            Step 4.2 $T_{sv} \leftarrow Slope(valleys_j, valleys_{j+1})$
            Step 4.3 $T_{dp} \leftarrow Disparity(peaks_j, peaks_{j+1})$
            Step 4.4 $T_{dv} \leftarrow Disparity(valleys_j, valleys_{j+1})$
        $\tau_i \leftarrow \{T_{sp_i}, T_{sv_i}, T_{dp_i}, T_{dv_i}\}$
    Step 5:
    for $i \leftarrow 0$ to $F_n$ do
        canonicalcoef$_i \leftarrow CCA(\tau_i)$
    Step 6:
    for $i \leftarrow 0$ to $F_n$ do
        coeffnew$_i \leftarrow CCA_{validate}((\tau_i, ..., \tau_{i+3}), (\tau_{i+3}, ..., \tau_{i+6}))$
        $variance_i \leftarrow CCA_{Variance}$(canonicalcoef$_i$, coeffnew$_i$)
    Step 7: BP$\leftarrow$ $peaks(variance_{sp}) \cup peaks(variance_{sv}) \cup peaks(variance_{dp}) \cup$
        $peaks(variance_{dv})$
    return BP
```

4.4 Experimental setup

The algorithms were implemented using Python platform. The CCA implementation that is available in Pyrcca [58] library was used in the algorithm. The data used in present work is English digits belong to the Indian accent. The speakers belong to different regions (states) in India. They include male and female speakers. We used 50 speaker's data in the analysis. Each English digit was recorded 15 times for all speakers. The digits were recorded using the Cool Edit software with 16KHz sampling rate, mono channel and 16 bits resolution. The performance of the algorithm for different cases are discussed in the next section.

4.5 Results and analysis

In the present study, a set of trajectory features are considered to be useful after conducting experiments on various properties. The properties that were observed are shown in Table 4.1. Figure 4.4 gives an idea of the nature of these features. They were not used as part of feature set in the segmentation process. They are useful in understanding the characteristics of regions belonging to different phonetic units. Some observations are presented in each subsequent subsections separately. The analysis of the algorithm's nature for peaks and valleys are presented separately in subsequent subsections.

Table 4.1: Attributes used for analysis

S.No.	Attribute
1	Peak
2	Peak width
3	Peak position
4	Average difference between adjacent peak values
5	Average slope between adjacent peak values
6	Valley
7	Valley width
8	Valley position
9	Average difference between adjacent valley values
10	Average slope between adjacent valley values

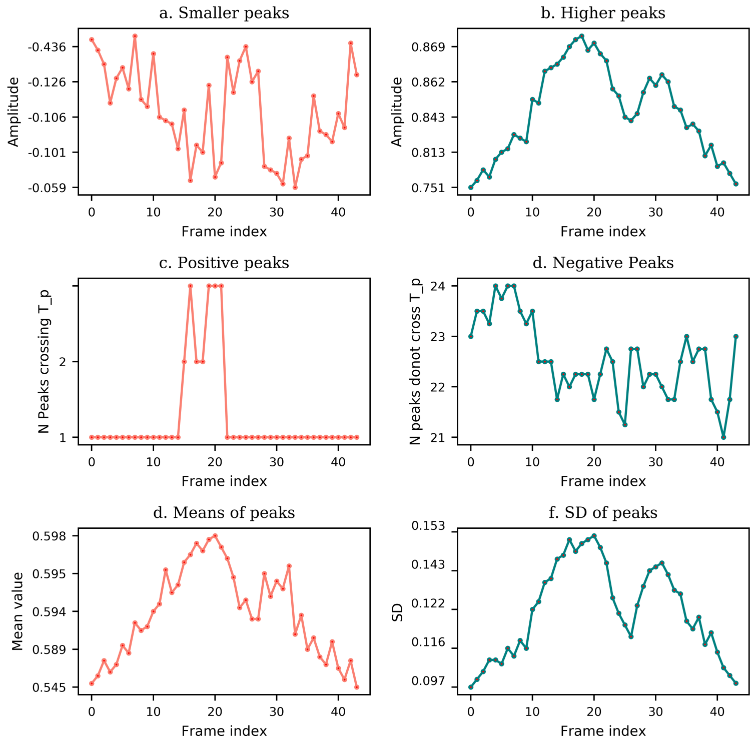


Figure 4.4: Peak statistics of the word "zero"

4.5.1 Peak attributes analysis

To understand meaningful cues from speech, an analysis of the nature of peaks in different classes of sounds like vowels, fricatives and stops are done. These clues are further used to find the boundaries of phonemes. It is helpful to know the regions where changes are occurring corresponding to the behaviour of attributes. Peaks can be classified into different types based on height and width. Vowels like /i/ and /e/ have regions with higher peaks and vowels /a/, /o/ and /u/ have wider peaks. Figure 4.2 shows different statistics of peaks. We can understand that the vowel regions have comparatively wider peaks than non-vowel regions. The analysis of slope feature vector can be done in two ways:

1. Slope between adjacent peaks in the same frame
2. Slope between peaks of adjacent frames

This attribute is used for understanding structural significance at phoneme boundaries. Slope between adjacent peaks in the same frame does not have much variations.The difference between frames belonging to the same phonetic unit is small. But it is observed that this value is more at the phoneme boundaries. Slope between peaks of vowel regions and non-vowel regions give enough variations that helps in understanding the boundary points. Figure 4.5 and Figure 4.6 show slope and disparity between peaks of adjacent frames for the words "Zero" to "Nine". It can be observed that the changes in the waveforms are evident so that structural clues can be captured by features. There is an interesting phenomena observed especially in vowel regions. There is a linear growth of the slope and disparity at the beginning of the vowel region and both start decaying at the middle part and continuing till the boundary is reached. This nature is observed both in intra-frame and inter-frame analysis. There is a sudden increase in the slope value at the boundaries of different phonemes. The average disparity between peaks within vowel region is more than non-vowel regions. Figure 4.2 shows the disparity between peaks for the word "Zero". We can observe that there are prominent changes at boundary frames. The distance between inter frame analysis is to understand the nature of the peak values with their neighbouring frames. This distance is more at the phoneme boundaries when compared to interior regions of phonemes. Anyhow this

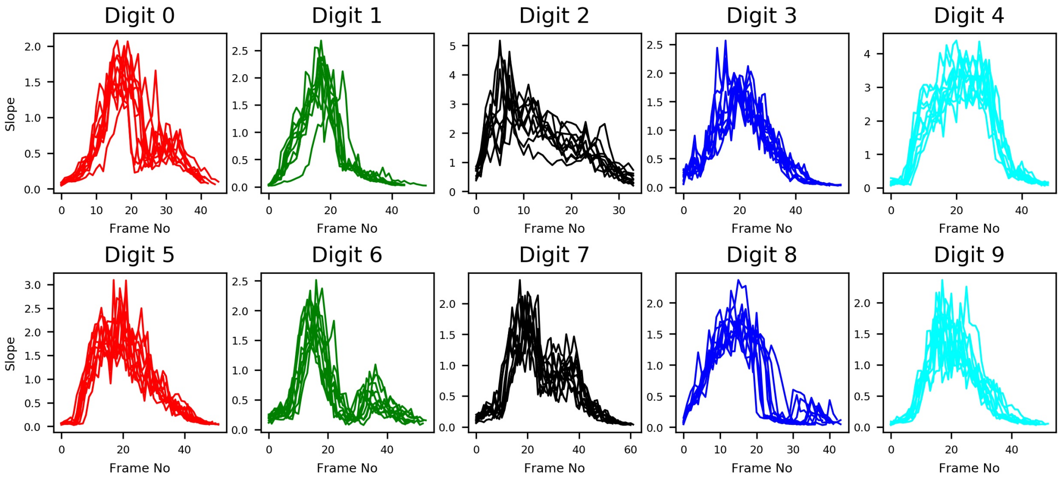


Figure 4.5: Slope between peaks of the words "Zero" to "Nine" for a speaker

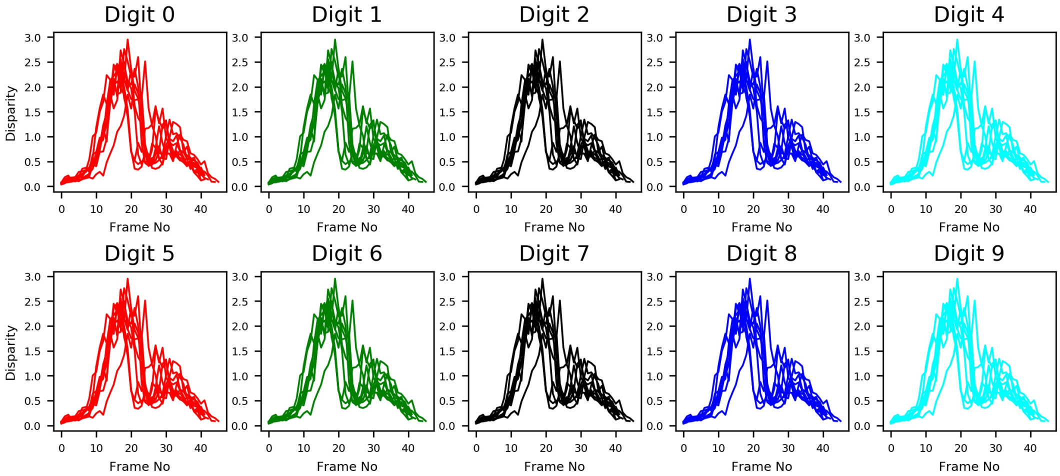


Figure 4.6: Disparity between peaks of the words "Zero" to "Nine" for a speaker

value is more in vowel regions like the intra frame difference. The difference between two frames is stable in the regions belonging to the same phoneme. Therefore it is inferred that intra frame difference can be used to identify the syllable boundaries whereas inter frame difference is useful in identifying phoneme boundaries. Figure 4.2 shows distance between peaks in adjacent frames for the word "Zero". It also shows that changes can be observed clearly at boundary frames of phoneme or syllable.

4.5.2 Valley attributes analysis

The second crucial feature of waveform in the framework is valley attributes. In this class, the nature of valleys was studied by understanding the properties of deeper valleys, shallow valleys, positive valleys, negative valleys, etc. Figure 4.7 shows the statistics of these attributes. The above mentioned properties with mean and standard deviation of valleys are shown in each sub figure. These graphs suggest that there is a temporal variation across the frames in these statistics which implies that the properties are significant for phoneme boundary analysis. We can understand variations in valleys for different segments of the speech sub-units.

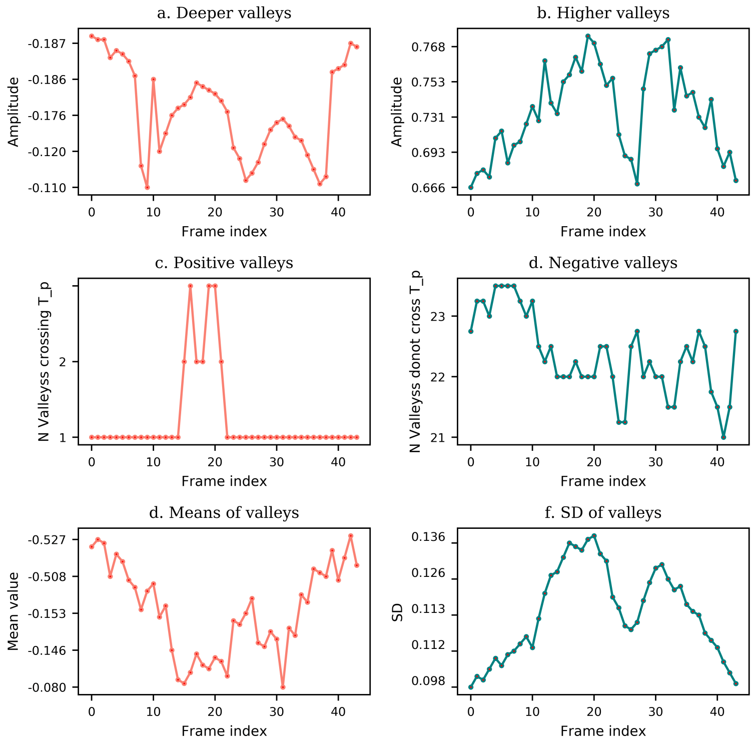


Figure 4.7: Valley statistics for the word "zero"

The observations from the analysis are listed below:

1. Deeper valleys and narrow valleys are found more in vowel regions than non-vowel regions.
2. Valleys in vowels are wide.
3. Standard deviation in vowel regions are comparatively higher than non-vowel regions.

These qualities mean that the structural variation can be exploited from valley features also. For example, vowels /i/ and /o/ have differences in the properties in terms of valleys. Vowel /i/ has deeper valleys compared to vowel /o/. It shows that there is more deviation between vowel and non-vowel regions. These statistics suggest that it is meaningful to use valley properties for understanding structural significance. The two properties Slope and disparity of the words "Zero" to "Nine" are shown in Figure 4.8, and Figure 4.9, respectively. We can see the structural consistency in different utterances of the same digit for a speaker.

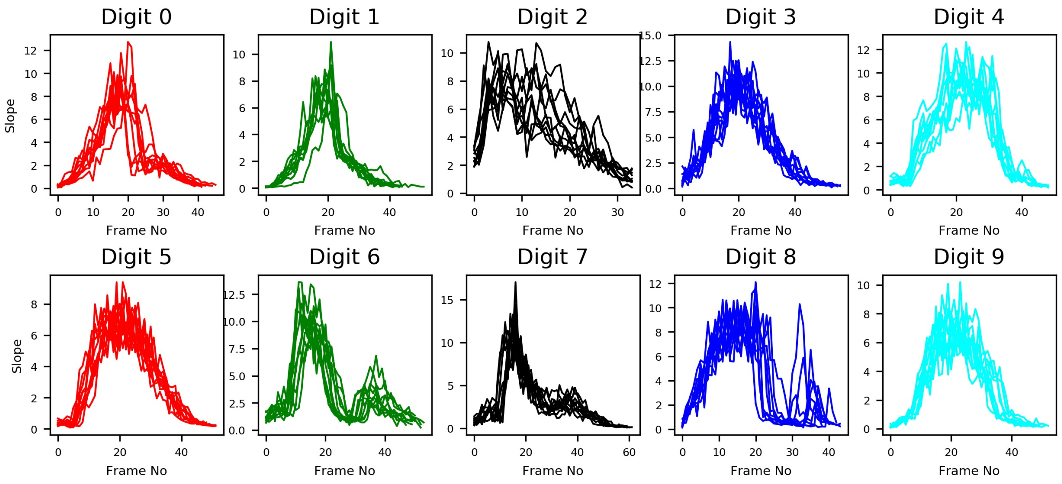


Figure 4.8: Slope between valleys of the words "Zero" to "Nine" for a speaker

4.5.3 Characteristics of the method in noisy conditions

The method was also evaluated in the presence of noise in input signals. Here, the white noise up to 20dB SNR was considered. Figure 4.10 shows a source speech signal along with the CCA coefficients of each feature vector. A comparison between Figure 4.3 and Figure 4.10 helps in understanding the nature of the algorithm in noisy signals. The first point to understand is that there is a variation in structure of same feature vectors. In this example, the disparity vector differs in variance of CCA coefficients. The noise presence makes the adjacent frames belonging to two different phonetic units much higher in their variation that is reflected in the CCA coefficients. The multi-view analysis enables the method to learn necessary clues from different vectors. Therefore, the failure of capturing

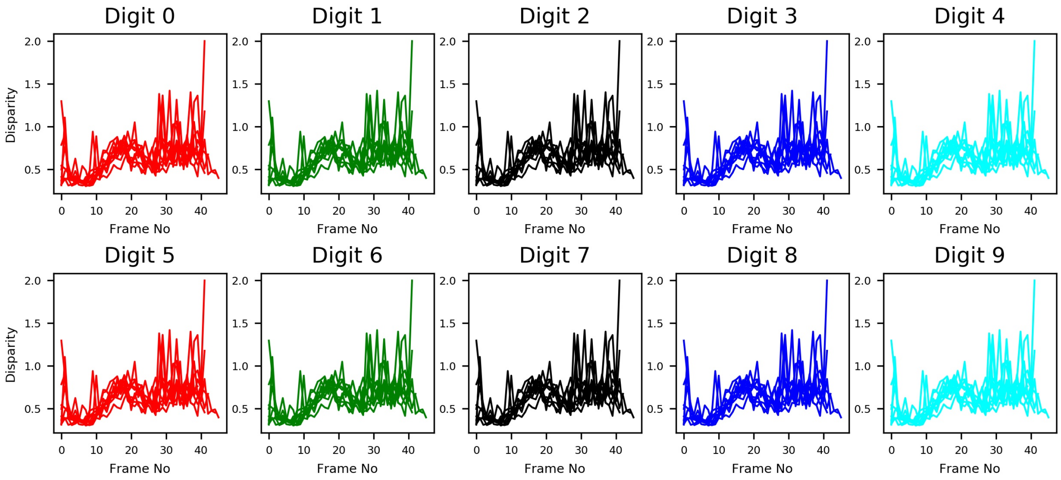


Figure 4.9: Disparity between valleys of the words "Zero" to "Nine" for a speaker

the boundary points in one case does not influence much in the final boundary points. So the results suggest that the proposed approach can be effective in noisy conditions also.

4.5.4 Performance of the algorithm

The proposed approach is successful in identifying the boundary points in 90% of the cases. The mis-identification of boundary points are influenced by speaker's characteristics in failure cases. This include accent, pauses between the phonetic units, etc. The time complexity of approach includes two major parts including feature extraction step and CCA. Time complexities of different steps are as follows:

1. Peak and valley computation: $\mathcal{O}(n)$.
2. Finding the trajectory properties need constant time $\mathcal{O}(1)$ for each elementary operation which constitutes a linear time complexity $\mathcal{O}(n)$ for n samples.
3. Lastly, CCA algorithm requires $\mathcal{O}(n^3)$ time complexity equivalent to eigen value decompositions method [59].

Therefore total time complexity of the approach works out to [$\mathcal{O}(n)$ + 4 x $\mathcal{O}(n)$ + 2 x $\mathcal{O}(n^3)$]. The run time requirement of the method is approximately 470 milli seconds. The method was tested on a system with the following configuration:

- Processor : i5 (3.20 GHz)
- Memory : 8 GB

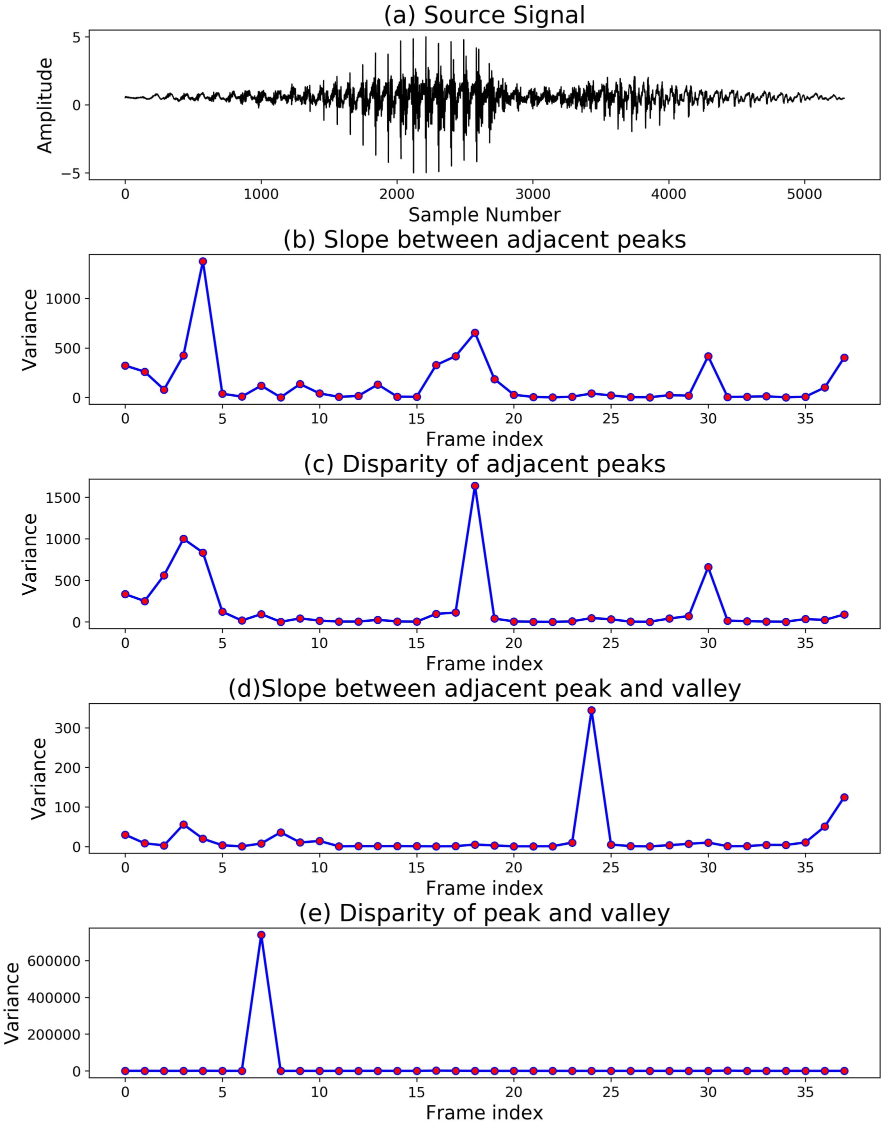


Figure 4.10: CCA variance of features for the word "Zero" in noise condition

4.6 Conclusions

In this chapter, a phoneme segmentation approach based on multi view geometrical features is proposed. The structural properties of speech trajectories are used to find the boundaries between phonetic units using the CCA method. The dissimilarities in geometrical features across a speech trajectory are used as parameters to identify boundary points. To prove the approach, Indian accented spoken English digits data was used in the experiments. The experiments gave reasonable results from which we can infer that the method is effective in identifying the boundary points. Since the approach does not require a training process, the requirement of large data sets are dispensed with. Also as the complexity of the method is reasonable, the run time is less and hence the method is suitable for low or zero resource languages. The data set has been shared in [*]

for the future use of the researchers. One of the problems with the approach discussed in this chapter is its inadequacy to find boundaries in a long sentence. Second issue in this approach is it requires multiple features. We investigated this issue further and found an approach that is capable of working with single set of features. This method is explained in Chapter 5.

[*] IITG DIGITS: https://drive.google.com/drive/folders/1px1p2p5QRNNvFvLJT9hgkA93N7$_{U}twzs$

Chapter 5

Phoneme Segmentation using Graph Eigen values

A phoneme segmentation algorithm based on geometrical structures of speech signals was discussed in the previous chapter. This technique works well for words and did not fare well in case of sentences. In this chapter, a graph based method for phoneme boundary detection is proposed. This method uses graph structures to analyze the wave forms. The approach is novel as it uses graph structure that has been proposed in [60]. Graph structure representation is entirely different to conventional speech signal analysis approaches. First, the data structure is explained with respect to a waveform. Next, the representation has been verified against the speech signals by studying the variation among different utterances using Graph Edit Distance. Using this, structural similarity among the phonetic segments is understood. Finally a segmentation algorithm is designed with graph eigen values. This chapter is organized as follows: An overview of graph representation is given in Section 5.1. The motivation for using graph based approach for signal analysis is described in Section 5.2. The proposed framework is elaborated in Section 5.3. Experimental setup and results are discussed in Sections 5.8 and 5.9 respectively.

5.1 Graph structures as features for signal representation

Recent developments [61] in signal processing techniques have made it easy to represent and analyze time series data using graph structures. There are several advantages in using graphs for feature representation. Similarly, structural based methods [3] are becoming a significant way to understand speech signals. It is evident that the objective of these procedures is to bring together phonetic features and speech dynamics. The advantage of these methods is to make the systems adaptive to speaker variability. Even though several structural processing methods have been utilized to process complex data, graphs on the other hand are suitable to store spatial and temporal data.

In general, a graph signal can be defined as follows:

$$G = \{V, A\} \tag{5.1}$$

where :

V (SET OF VERTICES) $= v_i, \forall i \leq N$
A (SET OF WEIGHTED EDGES) $= (v_i, v_j), \forall i, j \leq N$
N IS THE NUMBER OF VERTICES

In a graph, the components of a signal are represented as nodes and the edges represent similarity or closeness between these nodes. Causality in time series represents the order in which the events have occurred [62]. This is an important property which gives an useful ordering of the acoustic events in a speech signal. Graphs can be used to establish this property by adding edges between two nodes [61]. These reasons make graph analysis an effective way to study the changes in a time series. The methods that are available for analyzing graph structures are presented in the next subsection.

5.2 Graph based methods for identifying changes in time series

The idea of graph based methods for analysing time series can be seen from change point detection algorithms. In this approach, each data point is represented as a node in a graph and the complete time series as sequence of sub graphs. The structural transitions of the sub graphs help to identify the changes in the time series. This can be accomplished in different ways. Finding the sparsest cut of a graph is found to be a useful method to study the changes [63]. To understand the sparse nature of a graph, the following definitions are useful. A cut of a graph is defined as a partition of a vertex set into two subsets. A sparsest cut in a graph contains minimum number of edges among all the cuts. The sparsity of the cut is calculated using the spectral scan method. The approach proposed by Chen [64] uses a new parameter that is defined in Equation 5.2 .

$$R_G(t) = \sum_{(i,j)\epsilon G} \left[I_{g_{i(t)} \neq g_{j(t)}}, g_i(t) = I_{i>t} \right] \tag{5.2}$$

where:

G is the similarity graph on observations $\{y_i\}$
$R_G(t)$ signifies the number of edges that are connected between sample points before time instant t and after the time instant t
I_x is an indicator function for any occurence of event x. It can take *true* or *false* values

He. et.al proposed K-Nearest Neighbors (K-NN) [65] as a test parameter that uses the number of graphs that have close proximity to a graph. In this method, each graph is treated as vertex and the edges are added between them. The vertex or subgraph with least number of edges is detected as the boundary point. Another variation of K-NN is proposed in [66]. Further, Gaussian graph signals [67] are signal processing techniques where the input data is processed as graph structures. This approach uses Cumulative Sum (CUSUM) in which the pre-event knowledge and post-event are used. To know post-changes, Generalized Likelihood Ratio (GLR) has been used. The methods that are discussed so far have not been applied in speech signal analysis. A method for phoneme segmentation based on graph structures is proposed in the next section.

5.3 Proposed Framework

This section discusses the proposed framework in detail. The central idea is to process a speech signal as a graph. Here, a set of graphs are used to represent the complete signal where each graph substitute a segment of speech. Finally, a series of graphs are formed to represent a signal. This sequence of graphs are analyzed to understand the relation between each pair of adjacent graphs and subsequently this information is used for phoneme boundary detection. In other words, the framework is a triplet $\langle \chi, \delta_G, \mu \rangle$, where:

- χ - graph mapping function
- δ_G - similarity function
- μ - boundary detection criteria

Initially, the graph structure that is required in the analysis is constructed using a graph mapping function. The function transforms the input speech signal into a series of graphs. This will be processed further with a similarity function to extract essential parameters to find the boundary points. To do this, a boundary detection criteria is used. Each step is explained elaborately in the subsequent sections. Section 5.4 explains graph mapping function and the graph structure has been verified with a study using Graph Edit Distance in Section 5.5. Section 5.6 describes the similarity function and Section 5.7 details the boundary detection algorithm.

5.4 Graph mapping function (χ)

First we define the basic components of the graph structure. In a segment of speech $S[n] = \{x_i, x_{i+1}, ..., x_n, \forall i \in \mathbb{N}, \forall i \geq 0\}$, let x_{i-1}, x_i and x_{i+1} be consecutive samples.

The primitives peak and valley are defined as follows:

1. **Peak:** s_i is said to be a peak if the relation $s_{i-1} < s_i > s_{i+1}$ holds $\forall i \in \mathbb{N}, i \geq 0$
2. **Valley:**s_i is said to be a valley if the relation $s_{i-1} > s_i < s_{i+1}$ holds $\forall i \in \mathbb{N}, i \geq 0$

Now the graph G in this context is a set of vertices and edges. The vertices or nodes in this graph are peaks and valleys in the speech signal. Each vertex label is defined based on the position of peak (valley) in the given speech segment. Each node is associated with a weight i.e. the height and depth of peak and valley respectively. The edges in this graph are of three types. They are:

1. (v_i, v_{i+1}) - edge between two adjacent vertices
2. (v_i, p_j) - edge between vertex and peak
3. (p_j, p_{j+1}) - edge between two adjacent peaks

For a speech signal $S[n] = \{s_i, s_{i+1}, ..., s_n, \forall i \in \mathbb{N}, i \geq 0\}$ where n is the number of segments with equal size, the function χ is defined in Equation 5.3.

$$\chi : S \rightarrow G \tag{5.3}$$

where:

- G is a set of graphs
- χ is an onto function. It maps each s_i to one of the elements in G.

Therefore, the graph mapping function transforms a set of speech segments into a series of graphs. There can be three different structures possible for any segment of the speech signal. The difference is to reflect the varied number of peaks (or valleys). That means the number of peaks and valleys may not be identical everywhere. So the graph can accommodate these changes accordingly with respect to the number of peaks and valleys. The 3 different cases mentioned can be as given below:

1. Number of peaks and valleys are same
2. Number of peaks is greater than number of valleys
3. Number of peaks is less than number of valleys

The major steps in the approach are summarized as follows:

1. The speech signal is segmented into different non-overlapped frames of 15 ms.
2. Each frame is represented as a graph with vertices as peaks and valleys.
3. Edges of this graph are added as follows:
    a) An edge is added between every adjacent vertices
    b) An edge is added between every adjacent peaks
    c) An edge is added between consecutive peaks and vertices

The detailed steps in graph construction are given in Algorithm 7. This process results in a structure similar to the Figure 5.1. The effectiveness of the proposed graph structure has to be proven before hand to develop a segmentation algorithm. Therefore, we studied the suitability of the graph structure which is explained in Section 5.5.

Algorithm 7: Graph construction

```
Input:
S[N]: Speech signal of size N samples
Output:
G[nframes]: Graphs for nframes
begin
    Divide S[N] into $nframes$
    for $i \leftarrow 0$ to $nframes$ do
        $Peaks_i \bigcup p_i$                          ▷ Find peaks for each segment
        $Valleys_i \bigcup v_i$                        ▷ Find valleys for each segment
    for $i \leftarrow 0$ to $nframes$ do
        $G_i \bigcup peaks_i$                          ▷ Add peak node to graph
        $G_i \bigcup valleys_i$                        ▷ Add valley node to graph
        $niters \leftarrow argmin(peaks_n, valleys_n)$
        for $j \leftarrow 0$ to $niters$ do
            $E_i \bigcup (v_i, v_{i+1})$               ▷ Add edge $(v_i, v_{i+1})$
            $E_i \bigcup (v_i, p_i)$                   ▷ Add edge $(v_i, p_i)$
            $E_i \bigcup (p_i, p_{i+1})$               ▷ Add edge $(p_i, p_{i+1})$
        if $Valleys_n > niters$ then
            for $k \leftarrow 0$ to $Valleys_n$ do
                $E_i \bigcup (v_k, v_{k+1})$           ▷ Add edge $(v_k, v_{k+1})$
        else
            for $k \leftarrow 0$ to $Peaks_n$ do
                $E_i \bigcup (p_k, p_{k+1})$           ▷ Add edge $(p_k, p_{k+1})$
```

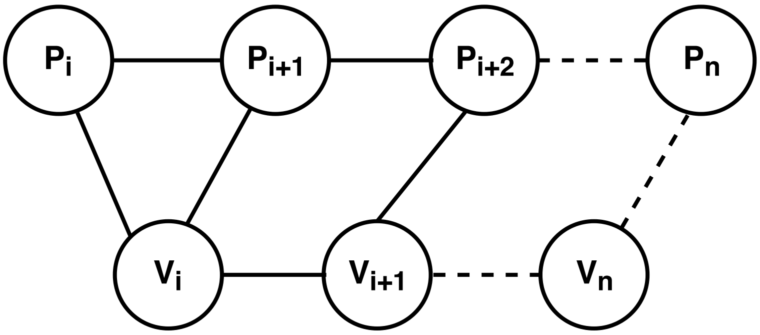


Figure 5.1: General structure of graph representation of a speech segment

5.5 Validation of the graph structure for phoneme segmentation

The appropriateness of graph structure can be understood by observing the patterns that are evident in a series of graphs. A pair of graphs (G_1, G_2) can be compared by a criterion called Graph Edit Distance (GED). GED can be defined as the effort needed to transform one graph to another graph in isomorphic form. There are numerous algorithms [68] available in literature to compute the graph edit distance. In general, the computation of GED considers all the possible ways of replacing the nodes of a source graph G_1 to transform it to the target graph G_2. The possible operations in this process would be substitution, insertion and deletion of nodes. The GED algorithm proposed by Reisen [69] [70] has been used as it requires polynomial time complexity. This algorithm works in two stages that are enlisted below.

1. Cost matrix representation
2. Cost matrix computation

$$\mathbf{C} = \left(\begin{array}{cccc|cccc} c_{11} & c_{12} & ... & c_{1m} & c_{1\epsilon} & \alpha & ... & \alpha \\ c_{21} & c_{22} & ... & c_{2m} & \alpha & c_{2\epsilon} & . & . \\ . & . & & & . & . & & . \\ . & . & & & . & . & & . \\ . & . & & & . & . & & \\ c_{n1} & c_{n2} & ... & c_{nm} & \alpha & ... & \alpha & c_{n\epsilon} \\ \hline c_{\epsilon 1} & \alpha & ... & \alpha & 0 & 0 & ... & 0 \\ \alpha & c_{\epsilon 2} & ... & . & 0 & & & \\ . & . & & & . & . & & . \\ . & . & & & . & . & & . \\ . & . & & & . & . & & . \\ \alpha & ... & \alpha & c_{\epsilon n} & 0 & ... & 0 & 0 \end{array} \right) \tag{5.4}$$

5.5.1 Cost matrix representation

The cost matrix of (G_i, G_j) is represented in a matrix of order $|n+m| \times |n+m|$, where n is the number of nodes in G_i and m is the number of nodes in G_j. The cost matrix is of the form given in Equation 5.4. This matrix C is partitioned into four parts $C_{00}, C_{01}, C_{10}, C_{11}$ to make provision for storing costs needed for node substitutions, insertions and deletions as mentioned earlier. C_{00} stores the costs for transforming G_1 to G_2 in terms of node substitutions and C_{01} is the deletion costs required. The node insertion costs are reflected in the diagonal of C_{10}. The 4^{th} partition C_{11} is kept as 0 since no costs are required for null operations.

5.5.2 Cost matrix computation

The second crucial step of computing cost matrix is obtained by using Munkre's algorithm [71]. For any two graphs G_1 and G_2, with vertex set V_1 and V_2 respectively, Munkre's algorithm works by mapping the nodes of V_1 to the nodes of V_2 such that the resulting cost is optimal. The initial costs of cost matrix C in Equation 5.4 are assigned by using Equations 5.5, 5.6 and 5.5.2. The entries in the matrix are modified using Munkre's assignment algorithm to get the final optimal cost of edit operations. The detailed algorithm for this process is found in [69].

$$c_{ij} = \begin{cases} 0 & \text{if edges are same} \\ 1 & \text{otherwise} \end{cases} \tag{5.5}$$

$$c_{\epsilon j} = \begin{cases} 1 & \text{if nodes are same} \\ \infty & \text{otherwise} \end{cases} \tag{5.6}$$

$$c_{i\epsilon} = \begin{cases} 1 & \text{if nodes are same} \\ \infty & \text{otherwise} \end{cases} \tag{5.7}$$

To illustrate the procedure, consider the waveform in Figure 5.2(a). After effective graph transformation, the edit distance between each pair of graphs for the adjacent frames is shown in Figure 5.2(b). It can be observed from the graph that there are different ranges of GEDs that constitute to different phonetic regions of the word. We can see clearly that there is a sudden change of GED that is associated with the change in the phonetic unit. GED patterns for different words are depicted in Figure A.2. In general, GED between vowel and non-vowel regions are higher comparatively with the other combinations. However, this property differ in case of long vowels where dissimilar structures are found in the same phoneme segment with less variability. The boundaries between vowels and non vowels are clear in nature whereas the unvoiced sounds like /k/ and /s/ in the word /six/, /e/ and /v/ in /seven/ are indistinguishable. The GEDs of the words (digits) "Zero" to "Nine" along with the wave forms are shown in Figure A.2. The study with the afore-mentioned observations suggest that the proposed structure can be useful in distinguishing phonetic units. Therefore, these structures can be used as representation in a segmentation procedure. We can infer from the study that the GED as a parameter representation may not be useful in finding the segmentation

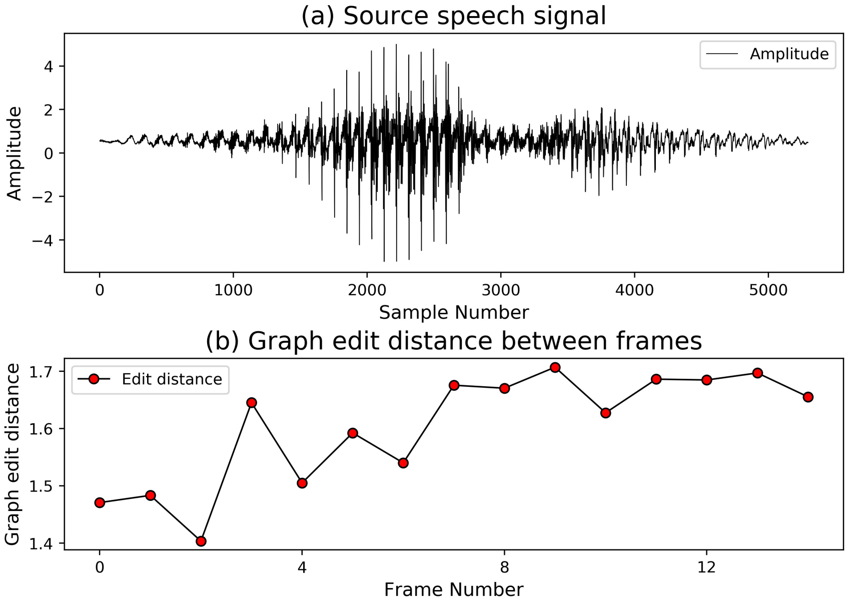


Figure 5.2: GED for an utterance of "Zero"

points since identical graph structures are possible for different phonetic units. This is caused due to the isomorphic structural transformation used in GED. Therefore, there is a need for a relevant feature representation for graph structure. In the next section, a feature representation based on graph eigen values is discussed that is used as similarity parameter in the segmentation procedure.

5.6 Graph Eigen values

In the previous section, the structural similarity between the graphs belonging to different segments of speech signal are shown. This proves the point that the graph can be used to represent the shape of a signal segment. The next step is to use this similarity as a measurement to actually understand the shape of a signal. The parameter that is to be used would be graph frequency. The frequency of vertices of a graph is retrieved and will be used as a parameter for finding the variation across the signal. The main component that is used in the approach is graph eigen value [72]. For a graph G, its adjacency matrix represented by $A(G)$ and Laplacian as $L(G)$, then the eigen values $\lambda(G)$ of graph G are equivalent to the eigen values of $L(G)$. The term $L(G)$ is given by Equation 5.8. Here $D(G)$ represents the diagonal matrix that consistis of degrees of G. The elements of $Ł(G)$ is a square matrix of size $n \times n$ and the elements are given by Equation 5.9. The condition for $\lambda(G)$ to be eigen value is if and only if $L(G)$ is singular and it can be expressed in terms of determinant given by Equation 5.10. The eigen values are computed by using Lanczos algorithm [73] [74] [75]. It depends on a parameter called Rayleigh quotient that is given by Equation 5.11. Here A is adjacent matrix and u, u^T are associated eigen vector and its transpose. A useful property of spectral theorem is that any vector that is orthogonal to A can diagonalize it.

$$L(G) = D(G) - A(G) \tag{5.8}$$

$$L(G)_{ij} = \begin{cases} degree_i, & \text{if i = j} \\ -A(G)_{ij}, & \text{if } i \neq j \end{cases} \tag{5.9}$$

$$|L(G) - \lambda(G)| = 0 \tag{5.10}$$

$$r(u) = \frac{u^T A u}{u^T u} \tag{5.11}$$

The summary of steps for computing eigen values for a speech signal segment is as follows:

1. Each speech signal is converted to its corresponding graph representation using graph mapping function (χ) that is discussed in Section 5.4.
2. The graphs generated in Step-1 are a set of adjacency matrices and referred to as A_G and its corresponding degree matrix is referred to as D_G.
3. Next, the Laplacian is computed using the Equations 5.8 and 5.9.
4. L_G is used to find the eigen values of the underlying graph of speech signal S_i. This procedure is also referred to as Lanczos algorithm.

An example o

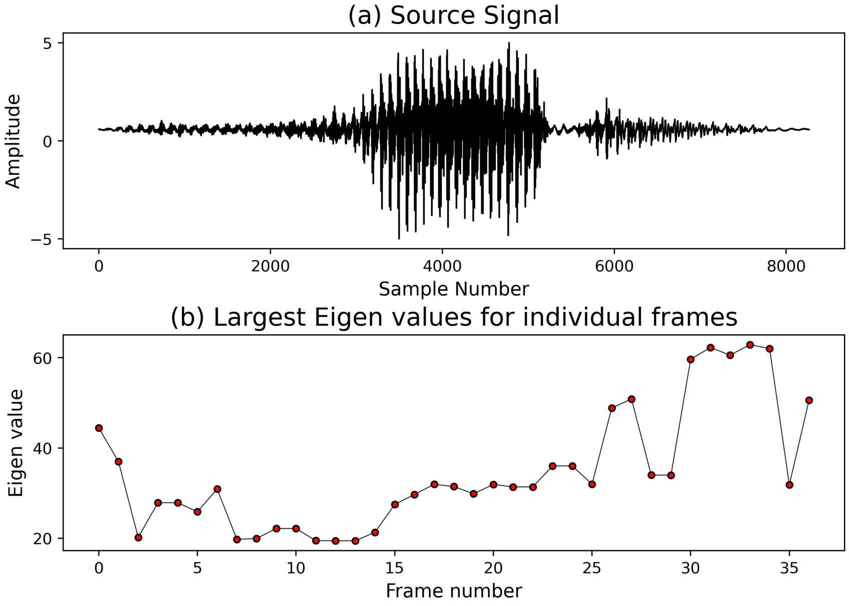


Figure 5.3: Largest graph eigen value for a speech utterance of the word "Zero"

5.7 Segmentation algorithm using graph eigen value

In Section 5.6, the complete procedure for computing graph eigen values was discussed. Eigen values are the central features that have been used in the approach. These features are to be processed to find the required temporal changes in the speech signal. The third component i.e. boundary detection criteria is used to accomplish this. The criteria that we use here internally employs Canonical Correlation Analysis (CCA) components. CCA finds the relationship between two variables by correlating them with a linear combination of variables that maximize the relationship. It can be used for two crucial reasons. First, co-variation between two variables can be derived with a small number of the linear combination. Second, the important features that can cause the co-variation can be understood. Graph eigen values corresponding to each speech segment is analyzed by CCA to find the crucial components represented as CCA components. These components are further analyzed to understand the variance between each pair of adjacent speech segments.
For each set of ′n′ features that represent ′k′ frames, the algorithm generates CCA mapping by projecting on each variable in the ′k′ values. The procedure works in two steps as follows:

1. CCA mapping of ′k′ variables is generated
2. Computation of variance between each set of ′k′ variables

In the first step, ′k′ elements are chosen from the available features. Each subsequent iterations slide towards right side for ′m′ positions. This is a simple sliding window approach with a window size of ′k′ and shift in ′m′ positions. For ′n′ elements, it requires $n - m$ iterations for each step to calculate CCA mapping of ′k′ variables. In the second stage, the correlation components that are calculated is used to understand the variance among them. This variance serves as an index for boundary points. The peaks in the variance signify the sudden changes in the speech signal. The steps in the segmentation algorithm are listed as follows:

1. Divide the input signal into a set of frames with equal length
2. The speech segments generated in Step-1 are transformed to a series of graphs $G = g_0, g_1, g_2, ..., g_n$ where n is the number of speech frames
3. After successful transformation, each graph is used to find graph eigen values using the procedure discussed in Section 5.6
4. The feature vector that is given by Step 3 is further processed by the boundary detection algorithm that is given in Algorithm 8. The algorithm generates a set of boundary points designated as B which gives start or end points of the phonemes.

A CCA computation of the eigen values for an utterance of the word "Four" is shown in Figure 5.4. We can see the boundaries of spoken units at the peak points in Figure 5.4-(c).

In the next section, the details of the data set, libraries and tools are explained.

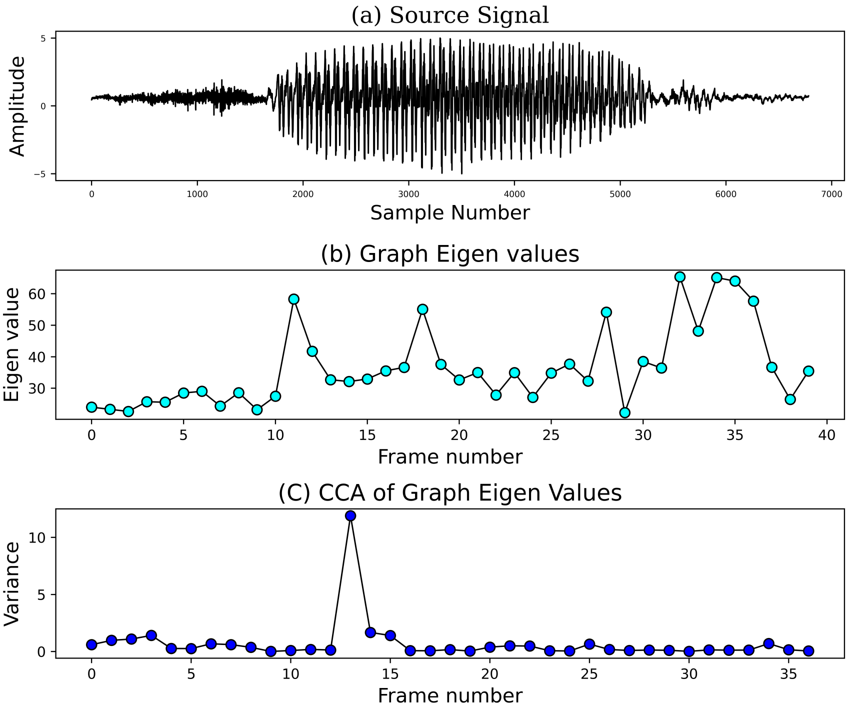


Figure 5.4: Boundary points of the word "Four"

Algorithm 8: Boundary detection algorithm

```
Input:
S[N]: Speech signal of size N samples
Output:
B[nsegments]: Boundaries of phonemes
begin
    Step 1: Divide S[N] into $nframes$
    Step 2: for $i \leftarrow 0$ to $nframes$ do
        $G_i \leftarrow$ Construct Graph($s_i$)          ▷ Construct graph for $s_i$ using Algorithm 7
    Step 3: Compute graph eigen values for each graph
    for $i \leftarrow 0$ to $length(G)$ do
        $L(g_i) \leftarrow D(g_i) - A(g_i)$          ▷ Laplacian of graph $g_i$
        $ge_i \leftarrow \frac{U^T(g_i)A(g_i)U(g_i)}{U^T(g_i)U(g_i)}$
    Step 4: Find variance between each pair of frames
    for $i \leftarrow 0$ to $length(G)$ do
        $cv_i \leftarrow CCA(ge_i, ge_{i+1})$
    for $j \leftarrow 0$ to $length(G)$ do
        $V_j \leftarrow Variance(cv_j, cv_{j+1})$
    for $k \leftarrow 0$ to $length(V)$ do
        if $v_k$ is a peak then
            $B_k \leftarrow v_k$
```

5.8 Experimental Evaluation

The environment includes the following components:

- A set of programs
- Libraries
- Dataset consisting of recorded digits

The programs used in the approach were implemented using Python programming language. Libraries include Networkx [76]. Scipy [77], and Pyrcca [58]. Networkx package accommodates built-in functions for manipulating complex graph structures. Using this library, sophisticated operations can be performed on graphs. Scipy is a standard Python library that supports routines for numeric operations. Finally, CCA implementation is available in Pyrcca. It supports regularized and kernel versions of CCA. In the present approach, regularized CCA has been used.

As the central objective of our research is to develop methods for low-resource languages, the data that we used belongs to native Indian English speakers. India is a country where the spoken English is influenced by the native Indian language. As a subcontinent, India has wide variety of languages in use. Each state has various dialects that differ in their speaking style. The data was collected from the people belonging to different regions that includes both male and female speakers of the age ranging from 20 to 26 years. There are 40 speakers involved in the recordings. Each digit was spoken 15 times thus a total of 6000 samples have been analyzed. The recordings were done using the Cool Edit software with 16KHz sampling rate, mono channel and 16 bits per sample. This data was used in one of the approaches published in [60]. The phoneme list in each word is given in Table 5.1. The analysis and the results obtained in the study are discussed in the next section.

5.9 Results and discussion

Graph structures are useful in representing the structural properties of acoustic events in an uttered word. This was proven with the help of pattern comparison using GED. Further-more, graph eigen values exhibit distinct properties that are essential to identify the abrupt changes that subsequently can be used for finding phoneme boundaries. The speech utterances were divided into equal sized frames with 160 samples (10 msec) without overlap. This frame size has been found to be an effective one compared to other frame sizes such as 100, 200 and 320. After a series of experiments with diverse segment sizes, finally the segment size was selected to continue with the experiments. Figure 5.5 shows largest eigen values of different words starting from "Zero" to "Nine". Each graph has 15 utterances of each word. Usually, the graph eigen values span in the range of 20 to

200 depends on the phonetic unit and its context. Especially vowel /u/ is superior to other vowels unlike vowel /o/ which is found to be inferior as seen in Figure 5.5-(a) and 5.5-(g) respectively.

Table 5.1: List of words

S.No	Word	Phonemes
1	Zero	/z/, /i/, /r/, /o/
2	One	/w/, /a/, /n/
3	Two	/t/, /u/
4	Three	/th/, /r/, /i/
5	Four	/f/, /o/, /r/
6	Five	/f/, /a/, /i/, /v/
7	Six	/s/,/i/, /k/,/s/
8	Seven	/s/, /e/, /v/, /e/, /n/
9	Eight	/e/, /i/, /t/
10	Nine	/n/, /a/, /i/, /n/

Essential variations to understand the boundary points are evident in the feature vectors in variance observed by CCA. However, there are few exceptions where changes are not noticeable. These are the silent regions that occur at the starting and ending of the words. The phonetic units with varying structures have dissimilar eigen values that influences the CCA variance. The speaker's speaking style can be a factor that decides occurrence of silence in an uttered word. Successively, it becomes difficult to identify the actual boundaries between two different phonemes as they share a common structure and the features have a low CCA variance. CCA variances of different words along with source speech signal and graph eigen values are depicted in Figures A.3 through A.12. In these figures, boundary points are identified at peak points of respective CCA variance. Table 5.2 shows segmentation accuracy of 10 different speakers. Successful detection rate varies from 74% to 84% with a comprehensive accuracy of 80% approximately.

5.10 Conclusions

In this chapter, a graph based method for analysing the phoneme boundaries has been proposed. The proposed graph structure was validated by comparing with Graph Edit Distance algorithm. After verifying the structures, a complete segmentation algorithm has been formulated by using graph eigen values and CCA method. The approach works by understanding the speech signal's shape through a sequence of graph structures. The proposed segmentation algorithm was evaluated in a low-resource data set of native Indian English speakers. The method has been found to be effective for the word data set with a success rate of 80% approximately.

Despite of its ability to find the phoneme boundaries in isolated words, it is not suitable for finding boundaries in a long sentence. This is due to the limitation of variance property that could not distinguish in a long span duration. To overcome the problem, a segmentation technique is proposed in Chapter 6 that uses fractal geometry to understand the properties of speech signal and capture the essential clues to segment phonemes in sentences.

Table 5.2: Segmentation accuracy for speakers 1 to 10 out of 40 speakers

Speaker Digit	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Zero	12	12	12	13	13	13	12	13	13	12
One	14	11	10	11	14	11	10	11	12	12
Two	15	12	13	11	10	09	13	12	10	10
Three	14	12	12	14	12	13	12	11	10	11
Four	13	11	11	13	12	12	11	11	12	12
Five	13	12	12	13	13	10	12	12	12	13
Six	11	12	13	12	13	13	13	12	13	14
Seven	10	10	12	10	14	10	12	14	13	12
Eight	12	12	13	12	11	10	13	13	14	12
Nine	13	10	11	12	13	10	11	12	11	14
Total Accuracy (%)	**84.7**	**76**	**79.3**	**80.6**	**83.3**	**74**	**79.3**	**80.7**	**80**	**81.3**

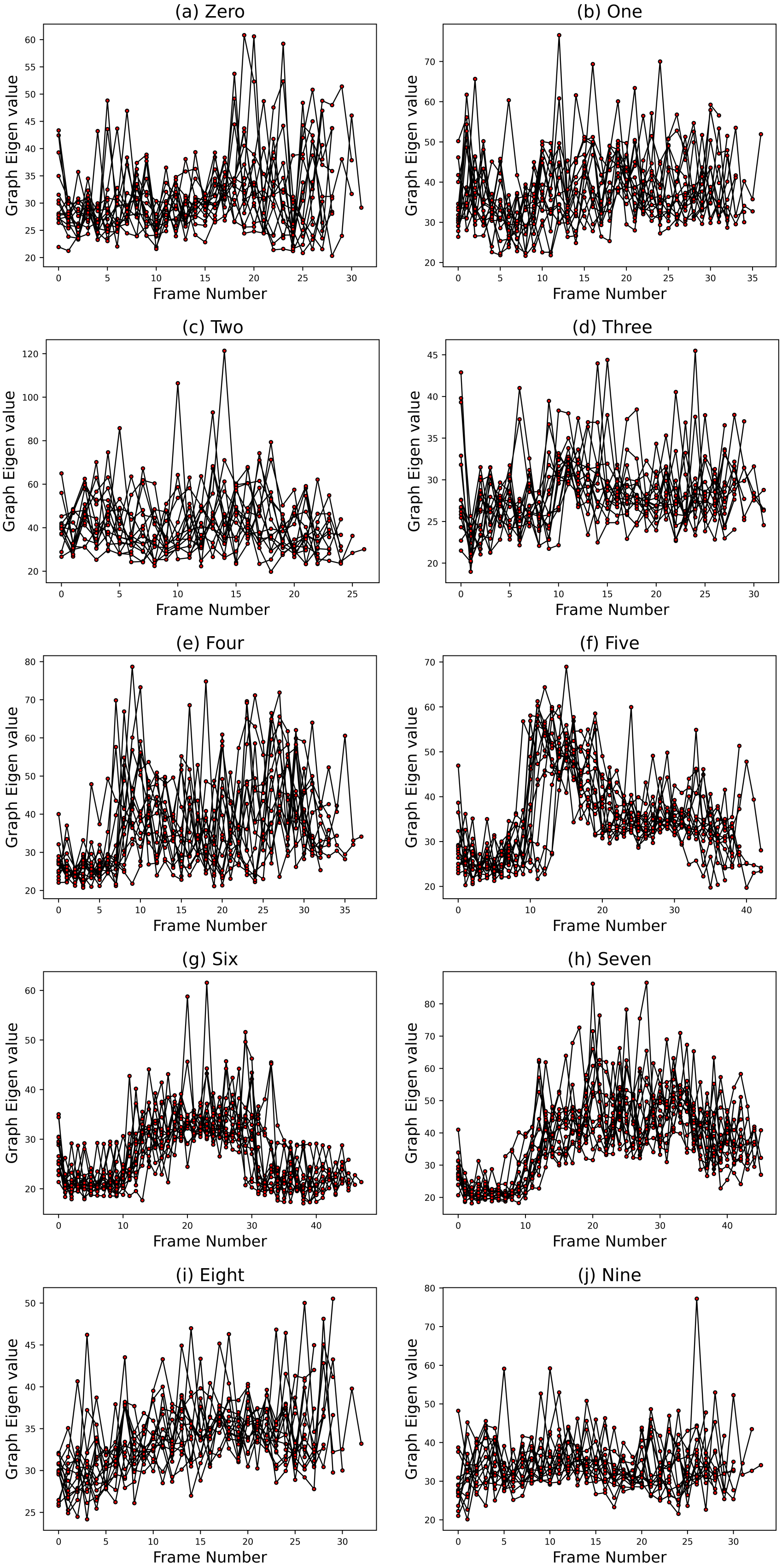


Figure 5.5: Graph eigen values of different utterances of the words "Zero" and "One"

Chapter 6

Fractal Features for Phoneme Boundary Analysis

The previous chapter discussed a graph based method for boundary detection. In this chapter a method for phoneme segmentation based on fractal analysis is presented. Fractal analysis is a well known approach in image processing and computer graphics. Fractals were introduced for understanding and measuring the properties of irregular shapes in nature. However this has not been thoroughly studied in the context of speech signals even though it has several useful properties that are suitable for signal analysis. Being a structural processing method, fractals have the capability to analyze the waveform properties. These properties are studied in this work and a segmentation procedure has been developed to identify phoneme boundaries in isolated word and sentences. In the study, the approach is found to be useful and computational efficient. The sections in the chapter are organized as follows: Section 6.1 gives a brief introduction to theoretical background. Section 6.2 explains the procedure used. Sections 6.3, 6.4 describes the environment used for experiments and results respectively. Section 6.5 gives concluding remarks of the chapter.

6.1 Fractals and speech signal analysis

Fractal geometry is an area that provides methods for analyzing irregular structures that exist in nature. This concept was first introduced by Mandelbrot [78] for characterizing objects that have irregular shapes. Fractals can be characterized by two properties: self-similarity and self-affinity. The property self-affinity belongs to only artificial shapes which are used in computer graphics. The natural objects are generally defined as self-similar objects. Self Similarity can be measured by a parameter called similarity-ratio and is defined in Equation 6.1.

$$r(N) = \frac{1}{N^{\frac{1}{D}}} \tag{6.1}$$

Equation 6.1 can be rewritten as

$$D = \frac{\log\left[\dfrac{N_l}{N_l+1}\right]}{\log\left[\dfrac{L_l+1}{L_l}\right]} \tag{6.2}$$

where:

l: construction level
N_l: number of segments in a curve
L_l: length of each segment

Fractal Dimension (FD) is another parameter that is used for characterizing fractal objects. This property helps in understanding the similarity between any two objects. A variety of approaches that are used to compute FD are described in the next subsection.

6.1.1 Approaches to compute fractal dimension

Often speech signal is represented as a waveform and it is considered as an open curve in two dimensions. A speech waveform exhibits both fractal characteristics and self-similarity. Analysing speech wave forms through fractals helps in understanding the shape of the underlying curve of a speech segment [78]. As discussed earlier, FD can be used to understand the properties of a signal shape. There are various methods available in literature for computing FD. Table 6.1 summarizes these approaches.

Table 6.1: Methods for computing fractal dimension

S. No.	Method	Parameters
1	BC	Number of grid squares covered by object
2	ASM	Number of speech samples crossing a certain threshold
3	IFS	FD obtained from a series of IFS parameters
4	Wavelet	Wavelet coefficients based on mother wavelet
5	PCA	Principal component matrix computed on trajectory state vector

Box counting method [79] is popular among the methods that are available. In this method, a grid of squares is used as a reference object. This is also known as structuring element. The FD computation of any object is simplified just by counting the number of squares that the object covers. The relation between FD and the number of objects (squares) is given by Equation 6.3.

$$N(n) \propto n^{-D} \tag{6.3}$$

where $N(n)$ is the number of objects whose linear dimension exceeds n and D is the dimension. But this method is proven to be invalid for speech signals due to conceptual inconsistency. An alternative method called Amplitude Scale Method (ASM) that is relevant for speech signal analysis has been proposed by Senevirathne et al. [80]. A crucial step in measuring FD for a speech signal is given by Equation 6.4.

$$D = \lim_{T_n \to 0} \frac{\ln[N(T_n)]}{\ln(\frac{1}{T_n})} \tag{6.4}$$

where T_n is a threshold and $N(T_n)$ is the number of threshold points considered. The threshold (T_n) is computed using different amplitude levels of the samples in a signal. The third method that is described next is Iterated Function System (IFS) in which an iterative procedure is used for extracting the IFS parameters [81]. A finite set of transformation functions are applied for parameter estimation. They are subsequently used in the process of FD computation. The transformation function is given in Equation 6.5.

$$W = \bigcup_{i=1}^{N} w_i(x,y) \tag{6.5}$$

where: $w_i(x,y) = \begin{vmatrix} a_i & 0 \\ c_i & d_i \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} e_i \\ f_i \end{vmatrix}$ and a_i, c_i, d_i, e_i, f_i are parameters generated from iterative IFS procedure and N is number of frames in the speech signal. Wavelet and PCA work by signal's decomposition and parameter reduction respectively. The approaches using wavelets and PCA are reported in [82], [83] and [84]. In the present approach, Mathematical Morphology (MM) has been used for fractal analysis. The reasons behind the usage of MM and the detailed steps for calculating the FD are discussed in the subsequent subsection.

6.1.2 Mathematical morphology for speech signal analysis

Morphology is a general concept which is used in areas like biology, linguistics, material science and signal processing. Speech processing involves the concepts of linguistics and signal processing. In linguistics, morphology is used for analysing the internal structures of different words in a language. Mathematical morphology [85] uses lattice theory and integral algebra. The key idea of morphological analysis is the understanding of the features of the regions in an image by filling those regions by different operators. There are various applications of morphology for both image processing [86] and speech signal processing.

The validity of the application of MM principles for speech signal analysis depends on the following two properties:

1. Partial ordering
2. Each subset should have a maximum and minimum

To check the above mentioned properties in the context of a speech signal, consider any speech signal $x[n]$ where $x[n] \subset \mathbb{R}$. Any set that is a subset of $\mathbb{R}$ satisfies the property partial ordering $\preceq$ by the relation $\leq$ for any pair of elements $x_i, x_j \ \forall i, j, \in \mathbb{N}$. The second property is satisfied in $x[n]$ because of the existence of peaks and valleys in the signal. They are the local maxima and local minima of any subset of elements in $x[n]$. Therefore MM principles can be applied for speech signal analysis.

Maragos proposed an approach for calculating the FD based on morphological covering [87]. This method was used effectively for phoneme recognition in combination with Mel Frequency Cepstral Coefficients (MFCCs). The present approach uses the principles of MM for the task of phoneme segmentation. The usage of fractals and MM in phoneme segmentation is not a new problem. These concepts have been used independently for phoneme boundary analysis. For example, Steinberg proposed a method to understand formant characteristics in speech spectrograms [88]. In his approach, Watershed transform (WT) was utilized for the segmentation task. This method takes motivation from natural phenomena occurring in geography. It was proven successful, but only for spectrograms. The input signal which has been used in the proposed approach is the raw waveform of speech. The steps in the proposed approach are explained in Section 6.2.

6.2 Proposed method for Phoneme boundary analysis

In this section, the modules in the proposed approach are detailed. The present framework consists of 3 major parts as follows:

1. Structuring element (G)
2. Area function $(\mathbb{A})$
3. Boundary detection criteria $()$

The modules mentioned above have different objectives in the phoneme segmentation framework. The structuring element (G) and area function $(\mathbb{A})$ were used in the work of Maragos and can be found in [87]. In his work, Maragos used these functions as basis for computing fractal dimension whereas the present approach uses it for calculating the area of a speech signal. The individual functions are explained in the following subsections.

6.2.1 Structuring element G

In mathematical morphology, operations are performed on the curve by associating each curve with a structuring element. This serves as a reference window or boundary for which the properties of the underlying curve are captured. There are different possible shapes (rectangle, triangle) which can be used as a reference object. The present approach uses a rectangle window. A window can be characterized by its width and height. The width of a structuring element of a sample s_i is defined based on its left and right neighbors s_{i-1}, s_{i+1} respectively $\forall i, 0 \leq i \leq N$. The height is defined as given by Equation 6.6.

$$h = \frac{Dynamic\ range\ of\ signal}{Number\ of\ samples} \tag{6.6}$$

The purpose of window element is as follows:

1. To apply the structuring element for each sample in a signal
2. To compute the area property
3. To control the movement of the structuring element once the area is known

The area can be computed using Algorithm 9. In the algorithm, Steps 4 through 7 are devoted to compute the structuring element.

6.2.2 Computing the area of the curve

The area for any given speech signal $S[n]$ is the cumulative area that is covered by all the samples $s_0, s_1, ..., s_k$. It is simply defined as the difference between dilation and erosion for a sample s_i and is given in Equation 6.7:

$$\mathbb{A} = \sum_{n=0}^{N} \Big[(S \oplus G_\varepsilon)[n] - (S \ominus G_\varepsilon)[n] \Big] \tag{6.7}$$

The terms $(S \oplus G_\varepsilon)[n]$ and $(S \ominus G_\varepsilon)[n]$ are dilation and erosion respectively. These operations are the primitive operators of mathematical morphology. For an input signal S, and structuring element G, the dilation and erosion are given in Equations 6.8 and 6.9 respectively.

$$\begin{aligned} S \oplus G[n] &= \max_{-1 \leq k \leq 1} \{S[n+k] + G[k]\}, \varepsilon = 1 \\ S \oplus G[n] &= (S \oplus G_\epsilon) \oplus G, \varepsilon \geq 2 \end{aligned} \tag{6.8}$$

$$\begin{aligned} S \ominus G[n] &= \min_{-1 \leq k \leq 1} \{S[n+k] - G[k]\}, \varepsilon = 1 \\ S \ominus G[n] &= (S \ominus G_{\varepsilon}) \ominus G, \varepsilon \geq 2 \end{aligned} \tag{6.9}$$

where S is the input signal and G is the structuring element which is given by the value **h** that was described in Subsection 6.2.1.

Algorithm 9: ComputeArea algorithm

```
Input:
s[i]: single frame of input signal
$\epsilon$: 1 to $\frac{nframes}{2}$
$h$: structuring element given by Equation 6.6
Output:
$X$: log areas of the given frame
$Y$: log areas of $\epsilon$
1  begin
2      Step 1:
3      for $i \leftarrow 1$ to $length(s[i])$ do
4          $a_1 \leftarrow s_{i-1}$
5          $a_2 \leftarrow s_i + h$
6          $a_3 \leftarrow s_{i+1}$
7          $a_4 \leftarrow s_i - h$
8          if $\epsilon = 1$ then
9              $dilation \leftarrow Argmax(a_1, a_2, a_3)$
10             $erosion \leftarrow Argmin(a_1, a_2, a_4)$
11         else
12             $dilation \leftarrow Argmax(a_1, a_2, a_3) + (\epsilon - 1) * h$
                 $erosion \leftarrow Argmin(a_1, a_2, a_4) - (\epsilon - 1) * h$
13         $area \leftarrow area + (dilation - erosion)$
14     Step 2: $area \leftarrow area + \epsilon^2$
15     Step 3: $X_n \leftarrow \log(area)$
```

By dilation operator, structuring element captures the features of a dilated image (signal), whereas erosion ensures the full contribution of the eroded speech segment. Another important parameter used in the procedure is ϵ (jumping factor). This is used for correlating different samples of the signal. Ideally this value can have a value in range of $1..n$, where n is the number of samples in a speech segment. For any sample s_i, $\epsilon = 1$ correlates s_i with its adjacent elements s_{i-1} and s_{i+1}. For $\epsilon = 2$, the proximity for correlation will be s_{i-2} to s_{i+2}. In the present approach, the highest value used for ϵ is $N_F/2$, where N_F is the number of speech segments available in the given input signal $S[n]$. As a whole, the function $\mathbb{A}$ defined in Equation 6.7 fills the curve of a speech segment which is subsequently treated as a topographical image. This is further used as the main feature for analysing the shape of the input signal. This process is described in Steps 3 to 13 of Algorithm 9 and depicted in Figure 6.1-(b). After filling the curve with area ($\mathbb{A}$) function, the next step is to identify the proper points using which a signal can be segmented. This process is decided by employing boundary detection criteria and is explained in the next subsection.

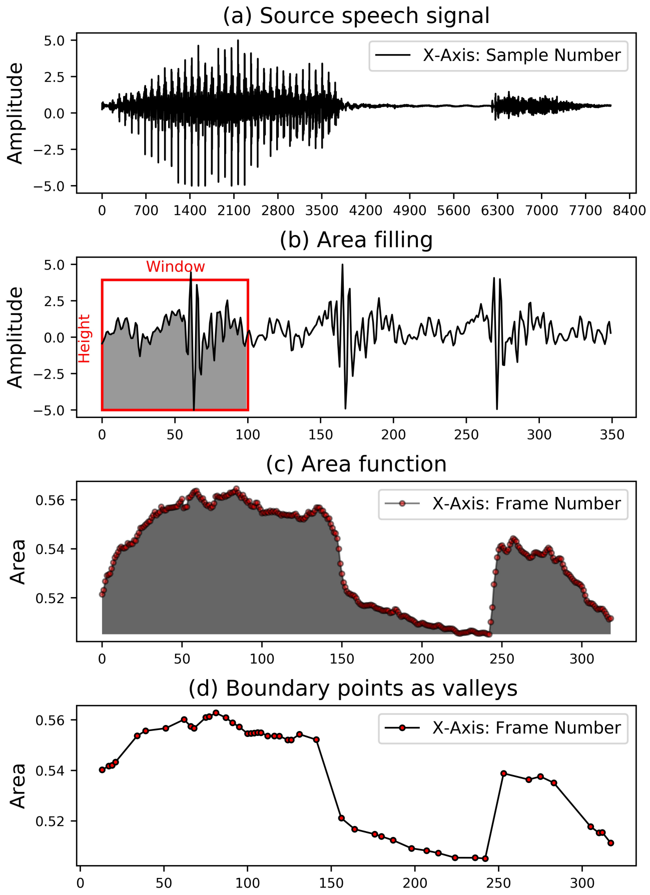


Figure 6.1: Steps in the boundary detection algorithm

6.2.3 Boundary detection criteria (λ)

Boundary detection criteria is used to find the exact phoneme boundary in the given input signal. It is the crucial and final step in identifying the exact frame in which the segmentation point exists. For this process, the range of area function is used as features. The distribution of the features gives an important pattern where the set of minimum points (valleys) and maximum points (peaks) occur at different positions. It is observed that the valley points in the waveform are the locations where changes in acoustic events are most prominent. Therefore segmentation criteria is defined on the nature of these valley points. A valley here is defined as follows:

Valley: In a segment of speech let s_{i-1}, s_i and s_{i+1} be consecutive samples, then s_i is said to be a valley if $s_{i-1} > s_i < s_{i+1}$ where $\forall i \in \mathbb{Z}$.

According to the above definition, there are multiple valleys found in the obtained features. But they are not the required segmentation points. Therefore these spurious valleys need to be discarded to get the exact boundaries. This is achieved by using the heuristics $H1$, $H2$ and $H3$.

H1 For any segment S_i and valley set $V_i \in S_i$, select a valley point v_{ij} such that $v_{ij} = Argmin(V_i)$.

H2 Let B_i be the set of valleys obtained by H1 and C_i is the cluster in which B_i belongs, then select only one element B_{ij} from B_i.

H3 Let $B = B_0, B_1, B_2, ..., B_n$ is the set of boundaries for the set $S = S_0, S_1, S_2, ..., S_n$ given by H2.

$$B_j \in \begin{cases} EB, & \text{iff } j \geq 3 \times i_{max}, \forall i_{max}, j \geq 1 \\ EB, & \text{iff } j \geq 3 \times i_{max} + 1 \ ,\forall i_{max}, j = 0 \end{cases} \tag{6.10}$$

where:

- i_{max} is the highest element of the set EB
- EB is the final set of segment points
- B_j is the new segment point

The valley points obtained by the heuristics H1, H2 and H3 are used as final segmentation points. The selection of valleys are illustrated in Figure 6.1-(d). The steps that have been illustrated constitutes the algorithm that is detailed in Algorithm 9.

6.3 Experimental setup

This section describes the environment that was used for conducting experiments. The algorithms were implemented using the Java programming language and Python. The English spoken digits of native Indian speakers have been used as data set. This contains 50 speakers belonging to different regions in India which includes males and females. 15 utterances of each digit are used. The digits were recorded using the CoolEdit software with 16 kHz sampling rate with mono channel. The data that was used in the experiments were normalized and DC component was removed. Each waveform is divided into a number of frames of fixed length. We used window size of 100 samples with an overlap of 20 samples. Experiments have been conducted on both clean and noisy speech signals. Apart from the above said data set, the algorithm has been tested on a small subset of TIMIT sentences data set. In the experiments, we analyzed 10 different sentences for 40 speakers which consists of different dialects. The results are discussed in the next section.

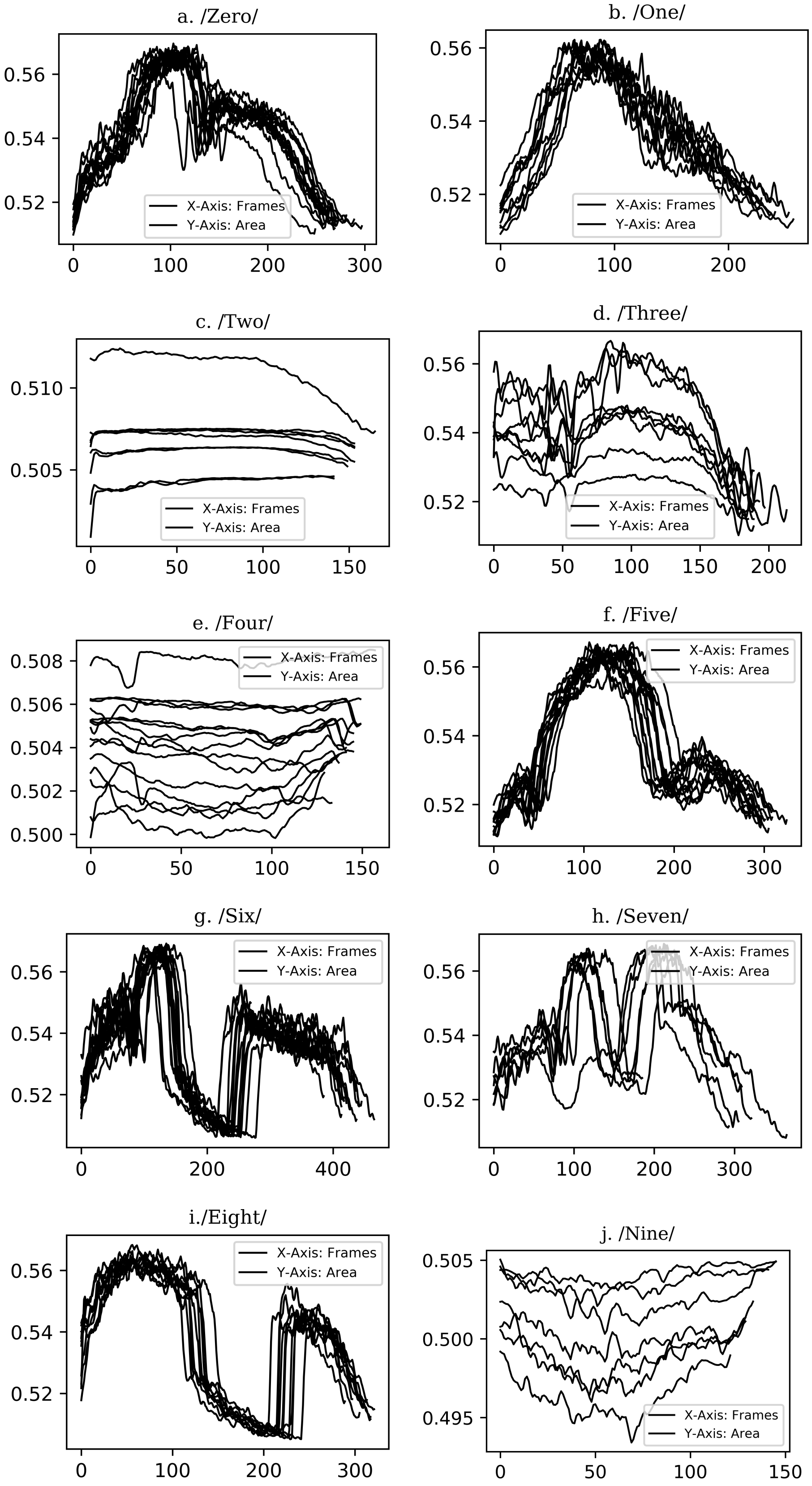


Figure 6.2: Area of different words ('Zero' to 'Nine')

6.4 Results and discussion

The proposed algorithm is effective in finding the phoneme boundaries using the area feature of the fractal. The area of the speech utterances for different digits belonging to a speaker are shown in Figure 6.2. There are different characteristics that have been observed by the function. The crucial points in observations can be discussed for different phonemic transitions. It gives different areas for the phonemic sounds like vowels, consonants, fricatives, etc. The behaviour of phonemes at transition points have been found to be remarkable. They provide clues for identifying the proper segmentation point. As a result, the method provides correct segment point in all the cases. The final boundary points that are calculated can be used for separating the spoken units rather that the individual phonemes. There are some issues that have been observed which can cause faulty segmentation. These observations are summarized in Table 6.2. As mentioned earlier, experiments were conducted on TIMIT sentence data set also. The graphs of two sentences spoken by two different speakers are shown in Figure 6.3 and Figure 6.4 respectively. In these figures, the area graph and boundary points of the algorithm for 2 different stages are shown. Area graph contains the possible boundary points that are found in the sentence. These points are further filtered to remove spurious boundaries and are shown in the third sub graph. The exact boundary points are at valley points in this graph. In the case of sentences, the approach is efficient in finding the word boundaries clearly. Even though it fails in few cases while detecting segmentation points for phonemes like /k/, /n/, /t/, /u/, /ly/, /Aa/ and /l/, the overall segmentation rate is reasonable in both the data sets. These failures can be characterized by the nature of data where the speaking style of speaker influences the boundaries of actual phonemes.

In the next subsection, the performance of the algorithm on noisy data is discussed.

6.4.1 Performance in noisy conditions

It is observed that the algorithm is less sensitive to the noise conditions. The reason is that the parameter Area remains unchanged with respect to the change of amplitude in the input signal. i.e. $\mathbb{A}(S[n]) = \mathbb{A}(S[n-n_0])$ where $\mathbb{A}(S[n])$ is the area of the original signal and $\mathbb{A}(S[n-n_0])$ is the area of the shifted signal. This is due to the nature of morphological erosion and dilation operations. These operations along with the structuring element G are invariant to constant shifts of amplitude in S[n] [85]. In the present method, we have used $log(\mathbb{A})$ as a parameter mentioned in Step-3 of Algorithm 9. So the changes in amplitude cannot have a substantial effect in this value that reflects on the overall shape of the feature curve. This can be observed in the graphs of the feature vectors. The plots of feature vectors and the boundary points detected for clean signals and noisy signals are shown in Figures 6.5-k, 6.5-l, 6.6-K and Figure 6.6-L respectively .

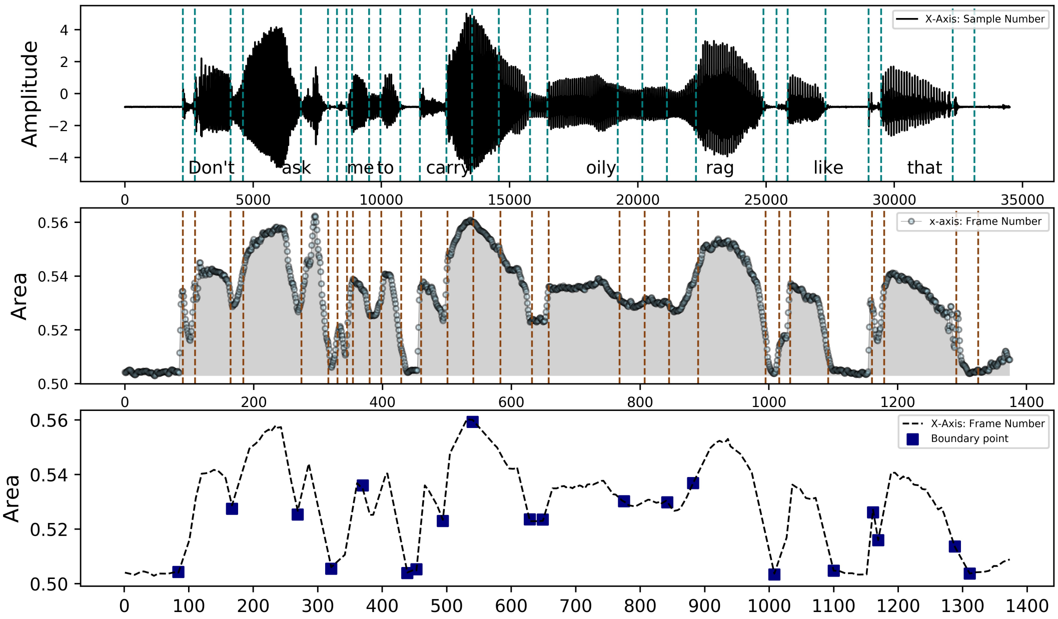


Figure 6.3: Boundary points for the sentence "*Don't ask me to carry an oily rag like that*"

We can see that there is very little difference in the locations detected as segmentation points.

6.4.2 Time complexity and comparison with existing methods

The observations lead to a conclusion that the present approach is not data driven and less complex in terms of run time. The complete segmentation procedure for a sentence can be completed in approximately 600 msec with time complexity $\mathcal{O}(n^3)$ for a speech signal of length n which includes the pre-processing step. System configuration in which the experiments have been conducted is as follows:

- Processor : i5 (3.20 GHz)
- Memory : 8 GB

After evaluating a set of experiments with words and sentences, it is found that the method gives reasonable performance in terms of finding the boundary points. Also, the performance of the proposed approach is compared with the existing methods. Table 6.3 summarizes the results. The proposed fractal segmentation algorithm works in 89% cases in Indian accented words and 87% cases for TIMIT sentences. The main advantage of the approach is it works in a single iteration by observing the properties of speech utterance and does not require a training process. This emphasises the usefulness of the approach.

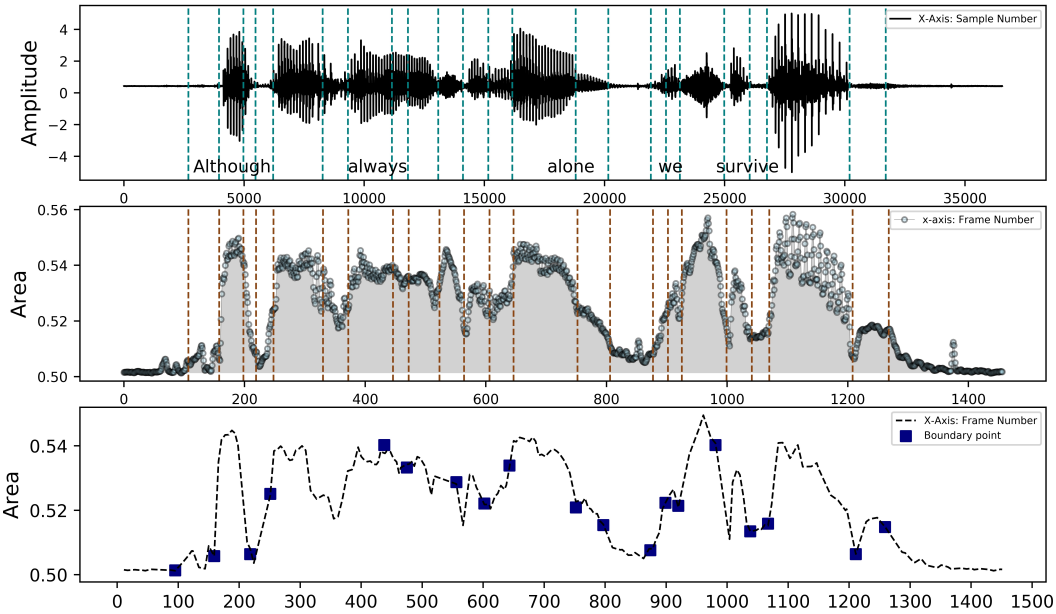


Figure 6.4: Boundary points for the sentence "*Although always alone, we survive*"

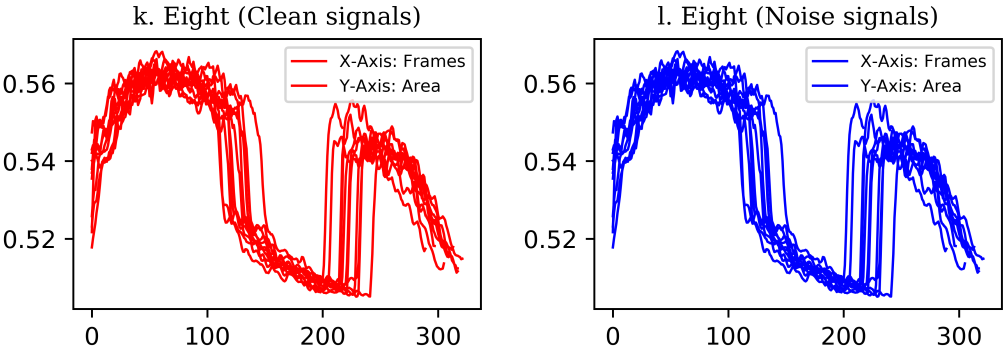


Figure 6.5: Comparison of clean and noisy signals (k, l)

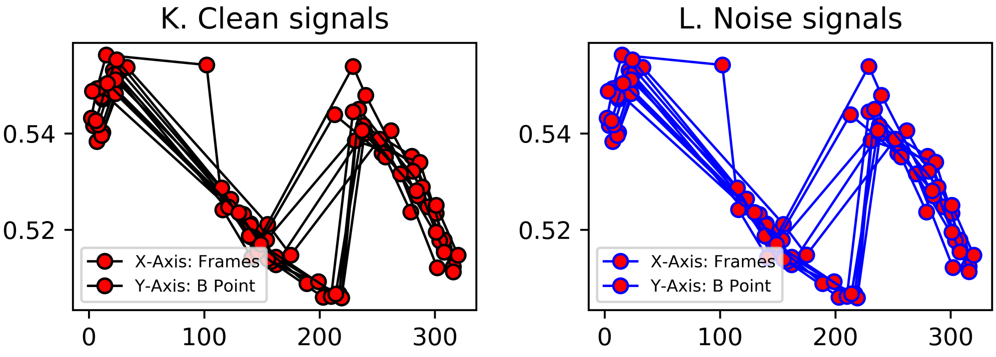


Figure 6.6: Boundary points of word /eight/ (K, L) for speaker Sp4

Table 6.2: Issues in segmentation

S. No.	Word	Remarks
1	Zero	Phonemes /Z/ and /i/ are combined in some cases
2	One	All units are segmented
3	Two	/u/ has some spurious points
4	Three	There is confusion in distinguishing /r/ and /i/
5	Four	Vowel /o/ has spurious boundaries within it
6	Five	All units are segmented
7	Six	/k/ and /s/ are sometimes combined as a single unit
8	Seven	/e/ and /v/ does not have clear boundaries
9	Eight	The transitions of /e/ and /i/ have confusion
10	Nine	Phonemes /i/ and /n/ are combined in few cases

Table 6.3: Performance comparison

S. No.	Method	Data set	Accuracy
1	LSTMs [89]	TIMIT	93 %
2	GMM [90]	TIMIT	81%
3	HMM [91]	Hindi sentences	73.7%
4	Proposed approach	Digits (Indian accent)	**88.7%**
5	Proposed approach	TIMIT	**87%**

6.5 Conclusions

In this chapter, a method for phoneme segmentation that uses fractal approach was proposed. The central feature that has been used is area under the waveform. This method works on the raw speech signal and gives frame level segment point in which phoneme boundary exists. To understand the performance, experiments were conducted on Spoken English digits of native Indian speakers and small set of TIMIT sentences data set. The detailed procedure with the results are discussed. It has been observed that the algorithm works accurately for 88.7% for the words and 87% for the TIMIT sentences. This correctness is comparable with the existing state of the art methods even though it is not superior in nature. The comparison with the current research methods is shown in Table 6.3. Therefore, the method can be effectively used in ASR systems to tokenize the events which can help in labelling them.

Chapter 7

Conclusions and Future work

This thesis addresses a new direction towards speech signal analysis and understanding based on structural processing. The approaches concentrate on the shape analysis of the speech signals. Each approach consists of a set of features and an analysis method. The proposed features can be classified into two categories as follows:

- Geometrical
- Graph based

Geometrical features are useful in understanding the properties of speech signals by their elementary geometrical properties whereas graph-based methods organize the primitives of wave forms so that patterns can be captured. It is found that both these classes of features are useful for the task of recognition and segmentation. In the first category, there are 3 attributes that are used and are enlisted below.

1. Peak attributes
2. Trajectory features computed with Fréchet distance
3. Tree structures

A trajectory based approach was discussed in Chapter 2. It uses two different features known as peak attributes and similarity features. The first one focuses on waveform primitives called peaks whereas the second one considers a trajectory representation for the entire utterance. The structural properties of the acoustic events that occur in different regions of spoken units are embodied in peak attributes. This nature makes them effective to characterize the spoken units that consists of single phonemes like vowels. The classification accuracy that was obtained with peak attributes was 75% for vowels and 58% for words.

The second type of trajectory parameters based on similarity measurement provided adequate clues that give a better classification for isolated words than the former one. The reason is that the temporal dynamics of an entire utterance is obtained in these features by which the knowledge of the complete structure is captured. This is not possible in peak attributes since they represent the segment-wise nature of individual events (or entities). These features were modeled using HMMs and it is proven to be efficient for classifying vowels and isolated words. This method resulted in 58% classification accuracy with digits in inter-speaker case similar to the former method whereas it successfully classified 90% in intra-speaker variability which is an improvement compared to 89% for peak attributes. This leads to an opinion that the classification capability of both the features are analogous.

Even though the approach is simple in terms of computational efficiency, we worked towards reducing the load of modeling without compromising the essence of the inherent structural properties. This goal to a new representation for waveform called Tree-structures was proposed in Chapter 3. Tree structure representation can be thought of a new means in characterization of spoken units. The approach considers the speech utterance at one go to form a holistic view and build a structural template. The representation arranges the primary entities of a waveform in a tree structure such that a pattern can be found. The final structure is subsequently used for classifying the phonetic units. The adequacy of the structures is proven with comparison using Zhang and Shasha Tree edit distance algorithm. A study was conducted by observing the pattern matching between a pair of trees that covers all the combinations of vowels. In the study, we found that the trees provided distinct patterns. The classification that was achieved with tree matching is 75%. It covers both the intra-speaker and inter speaker cases.

In the classification experiments, 7500 samples that consists of English spoken vowels and digits were used. There are 5 vowels and 10 digits uttered 15 times each. The data was recorded by 50 speakers belonging to different regions of India. People in each region speak different accents. Besides their suitability for recognizing the phonetic units, structural properties have exciting features to detect the boundaries between different phonetic regions. The geometrical features have been used for analyzing the phoneme boundaries in different contexts like words and sentences. To address this problem, three classes of features are proposed in the thesis.

1. Multi-way trajectory features
2. Graph-based features
3. Fractal features

The principles of structural components have been studied in various ways by using trivial geometrical features of peaks and valleys. The first approach was explained in Chapter 4. It uses multiple sets of features and the variation in the structure was analysed using a correlation analysis technique called CCA. The procedure was successful in identifying the boundaries between spoken units in a speech utterance. The issue with this approach is its requirement of multiple features since each kind of feature set captures different type of information. The problem of segmentation was further studied using graph based approach which was described in Chapter 5. The graph-based representation gives a way to represent the structure of different segments in the speech signal. Each segment gives a structural representation for individual segments. The similarity in these structures are used to capture the changes so that a point can be identified where a phoneme starts

Table 7.1: Methods Summary

S. No.	Method	Problem	Database	Accuracy (%)
1	Peak attributes	Recognition	Vowels	96
2	Peak attributes	Recognition	Digits	89
3	Fréchet distance	Recognition	Digits	90
4	Tree structures	Recognition	Vowels	70
5	Trajectory features	Segmentation	Digits	90
6	Graph eigen values	Segmentation	Digits	80
7	Fractal features	Segmentation	Digits	89
8	Fractal features	Segmentation	TIMIT	87

or ends. To match the pattern of a sequence of graphs, a Graph Edit Distance (GED) measurement was used. This pattern matching shows that the graph representation is appropriate for capturing the dissimilarity between two frames. However the GED was not advisable in graph comparison over long utterances because of its computational complexity. Therefore, a graph eigen value was taken as criteria in place of GED for similarity matching. This parameter represents frequency of vertex connectivity in a graph. Using this as a measure of change, a complete segmentation was designed. The method was successful in segmenting the phonemes in a word, but it was not efficient in sentences. To overcome this issue, a fractal based approach was proposed in Chapter 6. Fractal approach understands structural similarity across the speech utterance by using fractal properties. The properties are computed with simple mathematical morphological operations. The method was studied for identifying segmentation boundaries in words and sentences. In the study, it is found that the approach is capable of finding the phoneme boundaries. The issues with the detailed results are discussed in the thesis. In summary, the success rate of the segmentation algorithm is found to be 89% for words and 87% for sentences. The segmentation algorithms were tested on two different data sets consisting of words and sentences. There are 6000 utterances of 10 different words in the word database. The data was recorded by 40 different speakers belonging to different regions of India with varied accents. The sentences is a subset of TIMIT database that consists of 400 sentences of 40 different speakers from 10 different dialects. The approaches discussed in the thesis with the respective performance are shown in Table 7.1.

The boundary detection methods proposed in the thesis can be commonly referred to as single-scan algorithms. This class of techniques work by examining the clues available in the respective input signal. The nature of the algorithm does not depend on multiple instances of the same spoken units also. This is beneficial mainly for low-resource languages where huge amount of data is not in place to support development of segmentation procedures. Other advantage is that the computational requirement for segmentation is reasonably low as it is in the range of 550 msec to 650 msec for the fractal approach. This makes techniques to be suitable for annotation and segmentation in low resource devices also.

7.1 Future research directions

The studies so far conducted with the approaches have brought new insights through which speech signal analysis can be improved further. They are enlisted below.

1. The tree structures take into account the order in which the peaks and valleys occur. This didn't involve the frequency entity that a segment of a spoken unit contains. Therefore, they can be enhanced further by accommodating additional information of peaks and valleys. One possible aspect is to use amplitude (or height) of the peaks. In addition to that, a generic representation for the trees that represent the same phonetic unit can be modeled. Immediate choice to do this is a Median tree representation that gives a universal structure to a set of trees.
2. The Fréchet distance features are not modeled accurately for varied accents and speaking styles. But the method is proven effective to hold the variability that occur at different phonetic contexts. Being a low-cost feature, they can be combined with trees to provide better solutions. In the new method, feature vector will be represented as a tree instead of a raw waveform. In the thesis, we considered the tree structures for vowels alone. On the other hand, the similarity patterns represented with Fréchet distance features have crucial temporal dynamics of multiple phonetic units. They can be investigated further for inventing new intuitions towards phonemes classification.
3. Graph eigen value is a versatile attribute that can be used not only for graph structures, but also for acquiring important elements from multiple features. This aspect can help to combine multiple features. Especially this is a useful tool to process multi-way geometrical features to form a single set. Finally, eigen value can capture the essential components so that the variation can be used for either phoneme classification or segmentation.
4. Graph-based analysis is a new way of understanding the speech signals. The basic structure stores waveform primitives and holds relationships among the entities in terms of ordering and causality. Alongside, properties like priority, amplitude can also be utilized.
5. In our work, only the segmentation problem has been addressed. Still it has a breadth to explore other problems with this. The patterns shown by Graph Edit Distance follow a common structural variation for a set of speech utterances with multiple phonemes. This can be used to extract useful information from a set of graphs by using existing graph learning algorithms.

Appendix

Appendix A

Appendix

A.1 CoolEdit Pro

Cool Edit Pro is an audio analysis tool that supports recording, editing and analyzing the sound recordings in a variety of ways. It supports two different representations temporal and frequency of waveforms. The sounds can be recorded with different sampling rates and varied number of channels. It works in Windows platform.

A.2 Networkx

Networkx is a library in Python platform by which a complex graph structures can be programmed. It supports operations such as creation, manipulation of various types of graphs that include digraphs and multi-graphs. Implementation of standard graph algorithms are available in this package. The nodes of a graph can be preliminary data types or complex data components like images and XML records. The Python methods that were used in the programs are as follows:

- *Graph.add_node()*: It is simple method that adds a node into the graph
- *Graph.add_edge()*: This method adds an edge with or without a weight
- *Networkx.adjacency_matrix()*: Returns the adjacency matrix representation of an underlying graph

A.3 Similarity measures

Similarity measures is a Python library that supports the quantification of the difference between two arbitrary curves. It consists of methods to support different operations such as DTW (Dynamic Time Warping) and Fréchet distance. The latest version available is *similaritymeasures 0.4.3*.

A.4 PyGSP

PyGSP is another Python library that provides methods and classes to process graph signals. It is built based on the Spectral graph theory. Few operations that it gives are enlisted below:

- Computing fourier basis
- Interpolation of signals
- Filters
- Graph eigen values

In the thesis work, two operations compute_fourier_basis() and estimate_lmax() were used. The former method calculates the full decomposition of eigen value whereas the later estimates the largest eigen value in an eigen vector.

A.5 Pyrcca

Pyrcca is a Python package used for estimating the CCA of a set of variables. The methods in this package enables to perform different steps of the process of correlation analysis. They include:

- pyrcca.train()
- pyrcca.compute_ev()
- pyrcca.validate()

After creating an object of CCA, the set of variables are to be used for generating correlation weights. This process is done by the train() method. The weights that were estimated are used for estimating variance component between the variables by using compute_ev() method. The last method validate() is generally used for evaluating the weights that were computed.

A.6 Hmmlearn

Hmmlearn gives an implementation of Hidden Markov Models. It facilitates model parameters such as transition probability matrix, initial probability matrix and training process. The crucial steps in the process of HMM training can be done with the help of the following methods:

- hmm.GaussianHMM() - build a HMM instance
- fit() - creates HMM
- predict() - obtain state sequence in HMM

A.7 Flowchart for the boundary detection algorithm using trajectory parameters

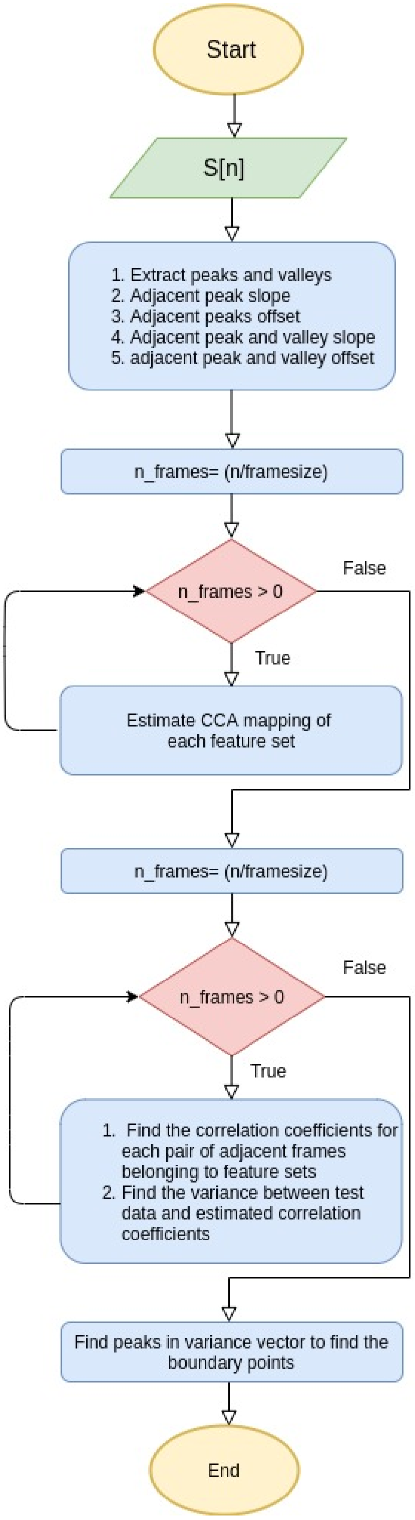

Start
S[n]
1. Extract peaks and valleys
2. Adjacent peak slope
3. Adjacent peaks offset
4. Adjacent peak and valley slope
5. adjacent peak and valley offset
n_frames= (n/framesize)
n_frames > 0
False
True
Estimate CCA mapping of each feature set
n_frames= (n/framesize)
n_frames > 0
False
True
1. Find the correlation coefficients for each pair of adjacent frames belonging to feature sets
2. Find the variance between test data and estimated correlation coefficients
Find peaks in variance vector to find the boundary points
End

Figure A.1: Flowchart for the boundary detection algorithm

A.8 Graph Edit Distance for words "Zero" to "Nine"

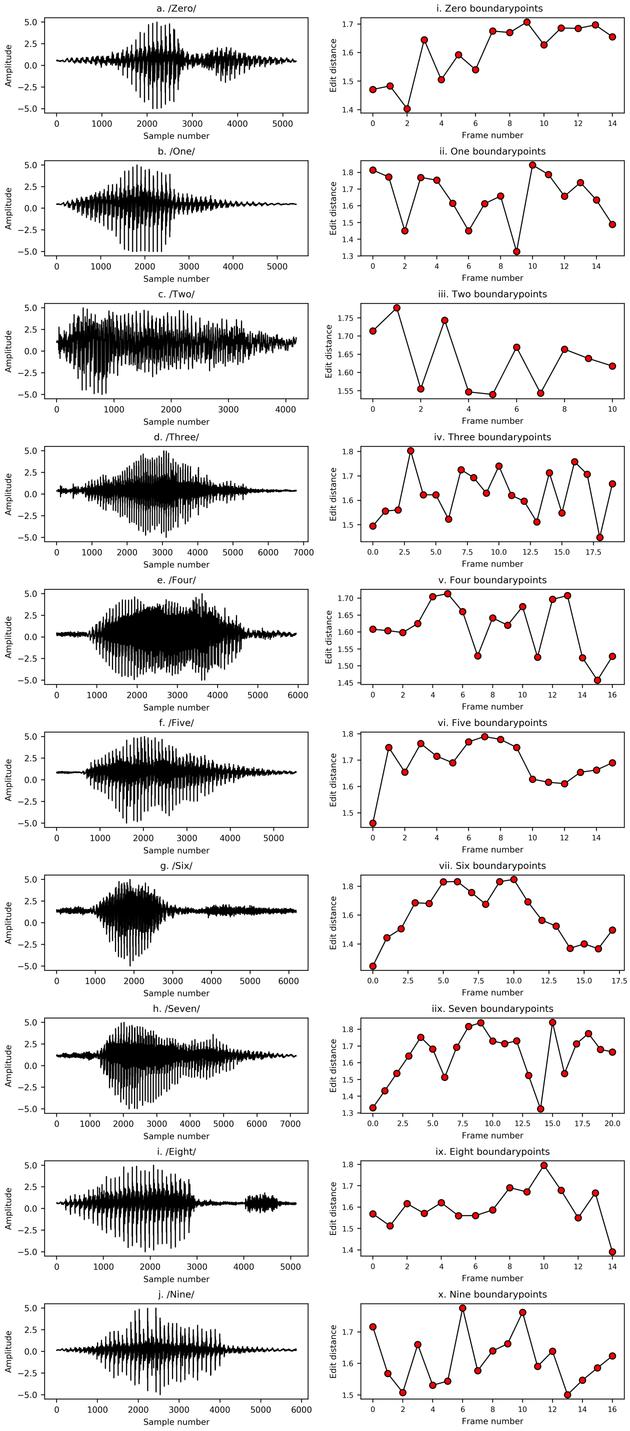


Figure A.2: Graph Edit Distance for words "Zero" to "Nine"

A.9 CCA variances of graph eigen values

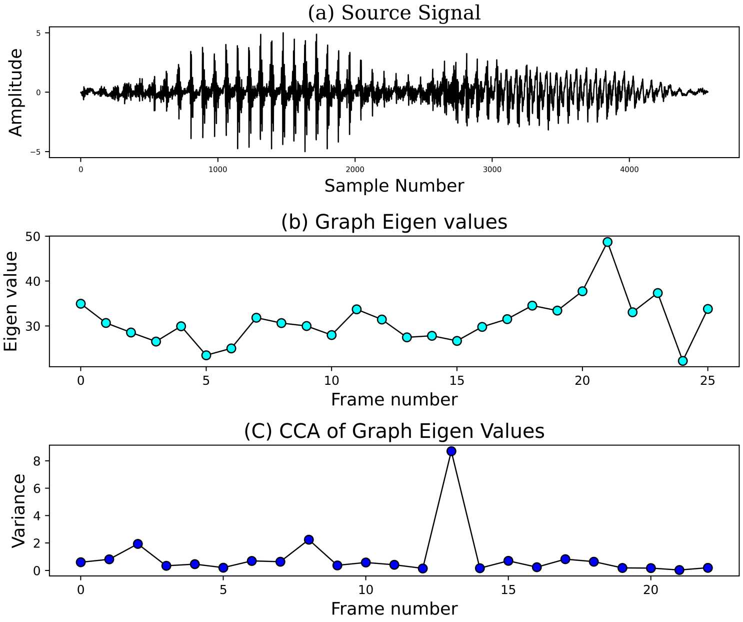


Figure A.3: CCA variance of graph eigen values for "Zero"

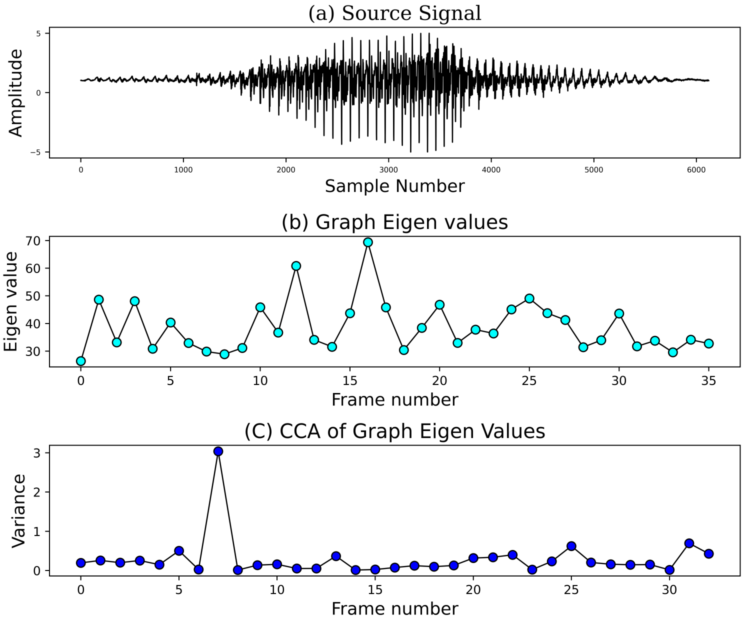


Figure A.4: CCA variance of graph eigen values for "One"

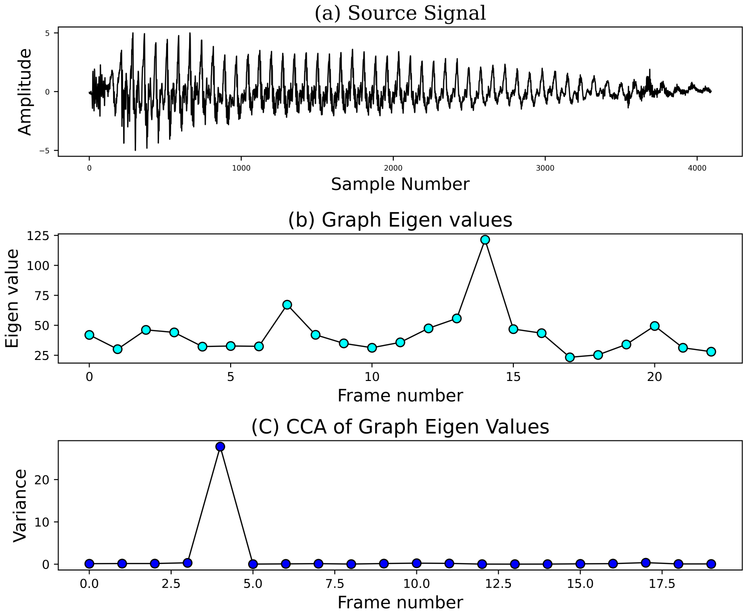


Figure A.5: CCA variance of graph eigen values for "Two"

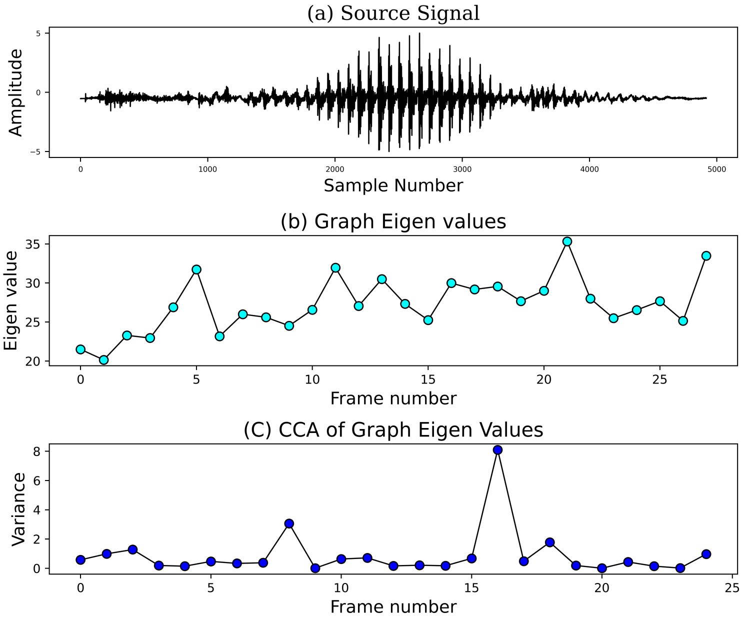


Figure A.6: CCA variance of graph eigen values for "Three"

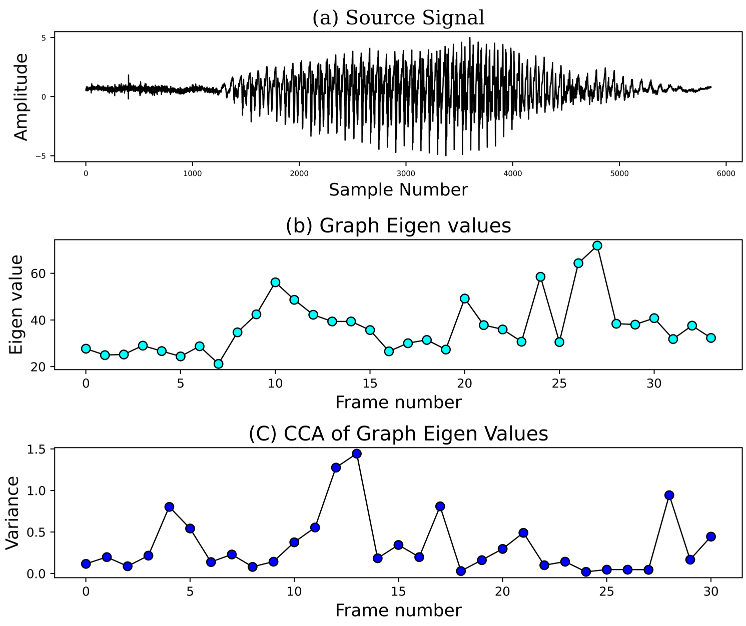


Figure A.7: CCA variance of graph eigen values for "Four"

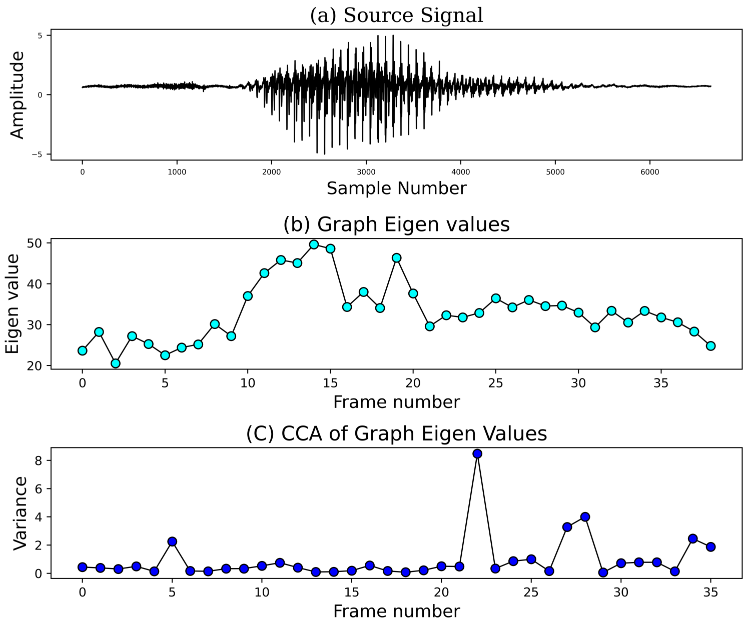


Figure A.8: CCA variance of graph eigen values for "Five"

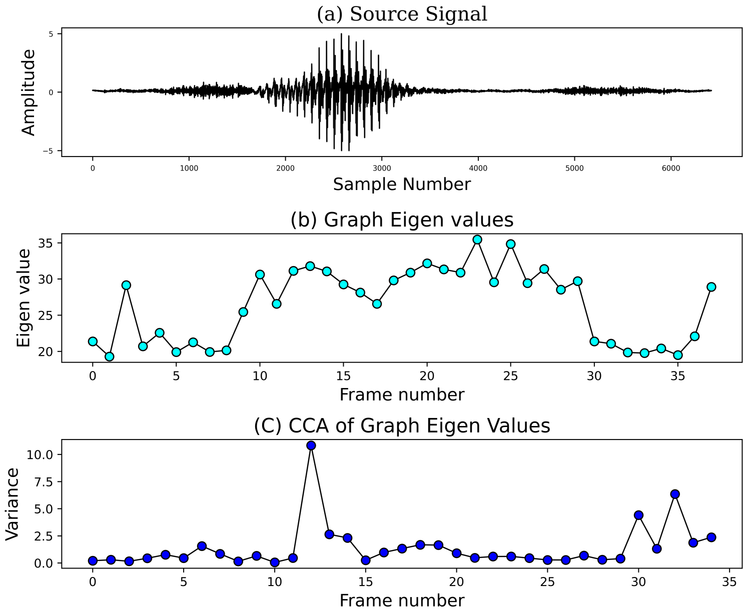


Figure A.9: CCA variance of graph eigen values for "Six"

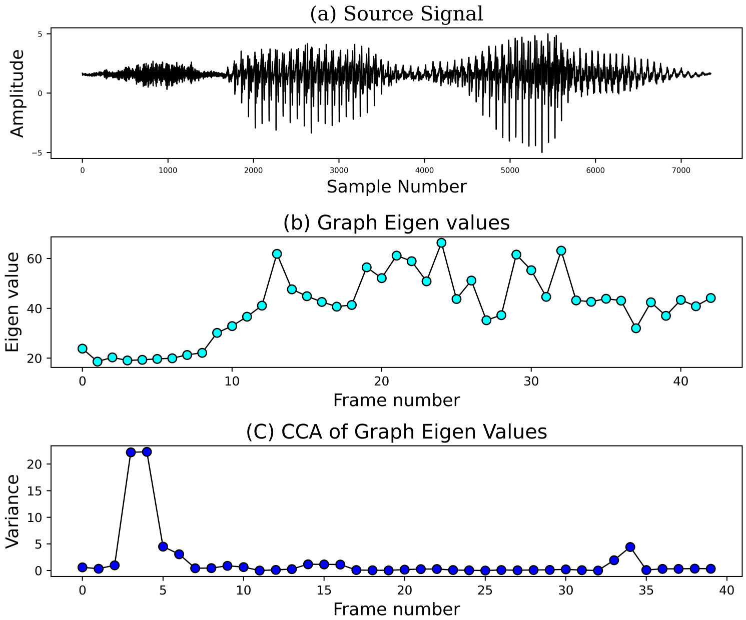


Figure A.10: CCA variance of graph eigen values for "Seven"

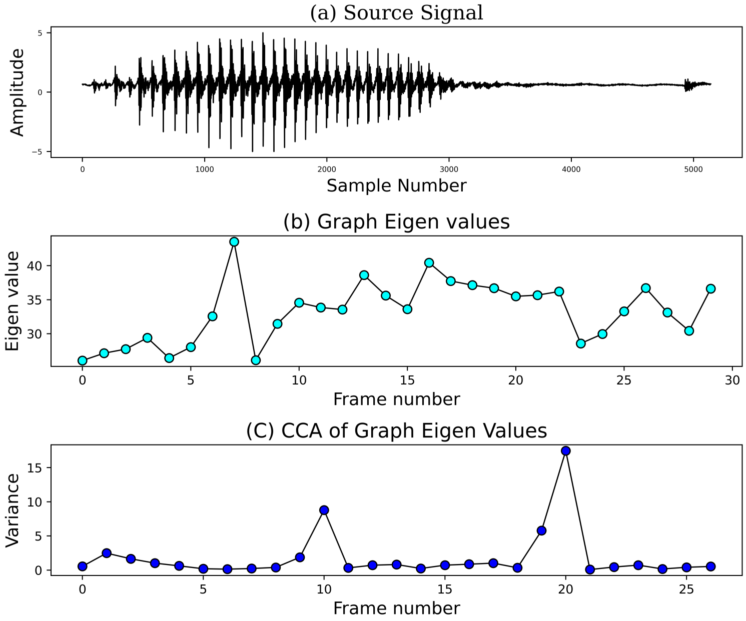


Figure A.11: CCA variance of graph eigen values for "Eight"

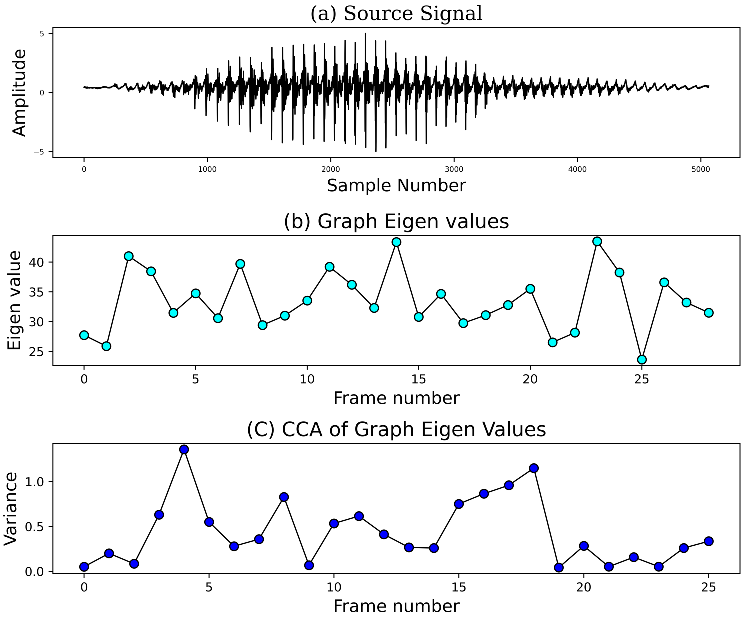


Figure A.12: CCA variance of graph eigen values for "Nine"

Bibliography

[1] N. Hubing and K. Yoo. 'Exploiting recursive parameter trajectories in speech analysis'. In: *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. Mar. 1992, 125–128 vol.1. DOI: 10.1109/ICASSP.1992.225956 (cited on page 7).

[2] K. E. Manjunath et al. 'Development of Consonant-Vowel Recognition Systems for Indian languages: Bengali and Odia'. In: *2013 Annual IEEE India Conference (INDICON)*. Dec. 2013, pp. 1–6. DOI: 10.1109/INDCON.2013.6726109 (cited on page 7).

[3] Li Deng and Helmer Strik. 'Structure-based and template-based automatic speech recognition - comparing parametric and non-parametric approaches'. In: *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007*. 2007, pp. 898–901 (cited on pages 8, 24, 55).

[4] N. Minematsu. 'Mathematical evidence of the acoustic universal structure in speech'. In: *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 1. Mar. 2005, I/889–I/892 Vol. 1. DOI: 10.1109/ICASSP.2005.1415257 (cited on pages 8, 18, 41).

[5] Patrick Garrigan and Philip J Kellman. 'Perceptual learning depends on perceptual constancy'. In: *Proceedings of the National Academy of Sciences* 105.6 (2008), pp. 2248–2253 (cited on page 8).

[6] Stuart Rosen. 'Temporal information in speech: acoustic, auditory and linguistic aspects'. In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 336.1278 (1992), pp. 367–373 (cited on page 8).

[7] Yifan Gong and J. -. Haton. 'Stochastic trajectory modeling for speech recognition'. In: *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. i. Apr. 1994, I/57–I/60 vol.1. DOI: 10.1109/ICASSP.1994.389356 (cited on pages 9, 10).

[8] H. Gish and K. Ng. 'A segmental speech model with applications to word spotting'. In: *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2. Apr. 1993, 447–450 vol.2. DOI: 10.1109/ICASSP.1993.319337 (cited on pages 9, 14).

[9] H. Gish and K. Ng. 'Parametric trajectory models for speech recognition'. In: *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*. Vol. 1. Oct. 1996, 466–469 vol.1. DOI: 10.1109/ICSLP.1996.607155 (cited on page 9).

[10] A. Kannan and M. Ostendorf. 'Adaptation of polynomial trajectory segment models for large vocabulary speech recognition'. In: *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 2. Apr. 1997, 1411–1414 vol.2. DOI: 10.1109/ICASSP.1997.596212 (cited on page 9).

[11] J. He and H. Leich. 'Speech trajectory recognition in SOFM by using Bayes theorem'. In: *Proceedings of ICSIPNN '94. International Conference on Speech, Image Processing and Neural Networks*. Apr. 1994, 109–112 vol.1. DOI: 10.1109/SIPNN.1994.344953 (cited on page 9).

[12] M. M. Thomson. 'Statistical modeling of speech feature vector trajectories based on a piecewise continuous mean path'. In: *1995 International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. May 1995, 361–364 vol.1. DOI: 10.1109/ICASSP.1995.479596 (cited on page 10).

[13] M. J. Russell and W. J. Holmes. 'Linear trajectory segmental HMMs'. In: *IEEE Signal Processing Letters* 4.3 (Mar. 1997), pp. 72–74. DOI: 10.1109/97.558642 (cited on pages 10, 14, 39).

[14] Young-Sun Yun and Yung-Hwan Oh. 'A segmental-feature HMM using parametric trajectory model'. In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*. Vol. 3. June 2000, 1249–1252 vol.3. DOI: 10.1109/ICASSP.2000.861802 (cited on pages 10, 14).

[15] Y. Minami et al. 'A recognition method with parametric trajectory synthesized using direct relations between static and dynamic feature vector time series'. In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. May 2002, pp. I–957–I–960. DOI: 10.1109/ICASSP.2002.5743952 (cited on page 10).

[16] Y. Minami et al. 'Recognition method with parametric trajectory generated from mixture distribution HMMs'. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*. Vol. 1. Apr. 2003, pp. I–I. DOI: 10.1109/ICASSP.2003.1198732 (cited on page 10).

[17] Yan Han, J. de Veth, and L. Boves. 'Trajectory clustering for automatic speech recognition'. In: *2005 13th European Signal Processing Conference*. Sept. 2005, pp. 1–4 (cited on page 13).

[18] P. Bhagath, M. Jain, and P. K. Das. 'Dynamic Speech Trajectory based Parameters for Low resource languages'. In: *2nd International conference on Machine Learning, Image Processing, Network Security and Data sciences 2020. Proceedings. (MIND '20)*. Vol. 1. June 2020, pp. I–I (cited on page 14).

[19] A. Kannan and M. Ostendorf. 'A comparison of constrained trajectory segment models for large vocabulary speech recognition'. In: *IEEE Transactions on Speech and Audio Processing* 6.3 (May 1998), pp. 303–306. DOI: 10.1109/89.668825 (cited on page 14).

[20] M. Z. Firouzmand and L. Girin. 'Perceptually weighted long term modeling of sinusoidal speech amplitude trajectories'. In: *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 1. Mar. 2005, I/369–I/372 Vol. 1. DOI: 10.1109/ICASSP.2005.1415127 (cited on page 15).

[21] Zelei Liu et al. 'A novel trajectory similarity–based approach for location prediction'. In: *International Journal of Distributed Sensor Networks* 12.11 (2016), p. 1550147716678426. DOI: 10.1177/1550147716678426 (cited on page 15).

[22] P. Xiao et al. 'Approximate Similarity Measurements on Multi-Attributes Trajectories Data'. In: *IEEE Access* 7 (2019), pp. 10905–10915. DOI: 10.1109/ACCESS.2018.2889475 (cited on page 15).

[23] Z. Lin et al. ‘A Semantic User Distance Metric Using GPS Trajectory Data’. In: *IEEE Access* 7 (2019), pp. 30185–30196. DOI: 10.1109/ACCESS.2019.2896577 (cited on page 16).

[24] H. Li et al. ‘Spatio-Temporal Vessel Trajectory Clustering Based on Data Mapping and Density’. In: *IEEE Access* 6 (2018), pp. 58939–58954. DOI: 10.1109/ACCESS.2018.2866364 (cited on pages 16, 40).

[25] Anne Driemel, Sariel Har-Peled, and Carola Wenk. ‘Approximating the Fréchet Distance for Realistic Curves in Near Linear Time’. In: *Discrete & Computational Geometry* 48.1 (July 2012), pp. 94–127. DOI: 10.1007/s00454-012-9402-z (cited on page 18).

[26] K. Bringmann. ‘Why Walking the Dog Takes Time: Frechet Distance Has No Strongly Subquadratic Algorithms Unless SETH Fails’. In: *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. Oct. 2014, pp. 661–670. DOI: 10.1109/FOCS.2014.76 (cited on page 19).

[27] Thomas Eiter and Heikki Mannila. *Computing discrete Fréchet distance*. Tech. rep. Citeseer, 1994 (cited on page 19).

[28] Lawrence R Rabiner. ‘A tutorial on hidden Markov models and selected applications in speech recognition’. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286 (cited on page 22).

[29] Charles F Jekel et al. ‘Similarity measures for identifying material parameters from hysteresis loops using inverse analysis’. In: *International Journal of Material Forming* (July 2018). DOI: 10.1007/s12289-018-1421-8 (cited on page 22).

[30] B. Zhao and T. Schultz. ‘Toward robust parametric trajectory segmental model for vowel recognition’. In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 4. May 2002, pp. IV–4165–IV–4165. DOI: 10.1109/ICASSP.2002.5745596 (cited on page 24).

[31] S. Chen et al. ‘Discrete Signal Processing on Graphs: Sampling Theory’. In: *IEEE Transactions on Signal Processing* 63.24 (Dec. 2015), pp. 6510–6523. DOI: 10.1109/TSP.2015.2469645 (cited on page 27).

[32] P. Bhagath and P. K. Das. ‘Characterization of Spoken English Vowels Using Tree Structures’. In: *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*. 2019, pp. 1758–1763 (cited on page 27).

[33] Bishnu S Atal. ‘Speech analysis and synthesis by linear prediction of the speech wave’. In: *The journal of the acoustical society of America* 47.1A (1970), pp. 65–65 (cited on page 28).

[34] Bishnu S Atal and Suzanne L Hanauer. ‘Speech analysis and synthesis by linear prediction of the speech wave’. In: *The journal of the acoustical society of America* 50.2B (1971), pp. 637–655 (cited on page 28).

[35] Steven Davis and Paul Mermelstein. ‘Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences’. In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366 (cited on page 28).

[36] R. W. Ehrich and J. P. Foith. ‘Representation of Random Waveforms by Relational Trees’. In: *IEEE Transactions on Computers* C-25.7 (July 1976), pp. 725–736. DOI: 10.1109/TC.1976.1674681 (cited on page 28).

[37] Y. C. Cheng and S. Y. Lu. ‘Waveform Correlation by Tree Matching’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-7.3 (May 1985), pp. 299–305. DOI: 10.1109/TPAMI.1985.4767658 (cited on page 28).

[38] S. Shaw and R. deFigueiredo. ‘Structural processing of waveforms as trees’. In: *ICASSP ’87. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 12. Apr. 1987, pp. 277–280. DOI: 10.1109/ICASSP.1987.1169679 (cited on pages 28, 37).

[39] GK Gupta and RC Joyce. ‘Using position extrema points to capture shape in on-line handwritten signature verification’. In: *Pattern Recognition* 40.10 (2007), pp. 2811–2817 (cited on page 28).

[40] Shun Ha Sylvia Wong, Anthony J Beaumont, et al. ‘A fuzzy decision tree-based duration model for Standard Yoruba text-to-speech synthesis’. In: *Computer Speech & Language* 21.2 (2007), pp. 325–349 (cited on page 28).

[41] d´túnjı A. d´jbı, Shun Ha Sylvia Wong, and Anthony J. Beaumont. ‘A modular holistic approach to prosody modelling for Standard Yorùbá speech synthesis’. In: *Computer Speech & Language* 22.1 (2008), pp. 39–68. DOI: 10.1016/j.csl.2007.05.002 (cited on page 28).

[42] M. H. Fisher and R. T. Ritchings. ‘Attributed relational tree approach to signal classification’. In: *IEE Proceedings - Radar, Sonar and Navigation* 141.6 (Dec. 1994), pp. 319–324. DOI: 10.1049/ip-rsn:19941522 (cited on page 28).

[43] Philip Bille. ‘A survey on tree edit distance and related problems’. In: *Theoretical computer science* 337.1-3 (2005), pp. 217–239 (cited on page 32).

[44] D. Shasha et al. ‘Exact and approximate algorithms for unordered tree matching’. In: *IEEE Transactions on Systems, Man, and Cybernetics* 24.4 (Apr. 1994), pp. 668–678. DOI: 10.1109/21.286387 (cited on page 33).

[45] Kaizhong Zhang and Dennis Shasha. ‘Simple fast algorithms for the editing distance between trees and related problems’. In: *SIAM journal on computing* 18.6 (1989), pp. 1245–1262 (cited on page 33).

[46] ‘Chapter 9 - Noise Analysis and Random Processes in the (t,f) Domain’. In: *Time-Frequency Signal Analysis and Processing (Second Edition)*. Ed. by Boualem Boashash. Second Edition. Oxford: Academic Press, 2016, pp. 521–573. DOI: https://doi.org/10.1016/B978-0-12-398499-9.00009-1 (cited on page 37).

[47] Lu Zhao et al. ‘Acoustic features of Mandarin monophthongs by Tibetan speakers’. In: *Asian Language Processing (IALP), 2014 International Conference on*. IEEE. 2014, pp. 147–150 (cited on page 37).

[48] Li Deng, Dong Yu, and Alex Acero. ‘Structured speech modeling’. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.5 (2006), pp. 1492–1504 (cited on page 37).

[49] O. Siohan and Yifan Gong. ‘A semi-continuous stochastic trajectory model for phoneme-based continuous speech recognition’. In: *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. Vol. 1. May 1996, 471–474 vol. 1. DOI: 10.1109/ICASSP.1996.541135 (cited on page 39).

[50] V. Mitra et al. ‘Articulatory trajectories for large-vocabulary speech recognition’. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. May 2013, pp. 7145–7149. DOI: 10.1109/ICASSP.2013.6639049 (cited on page 40).

[51] S. Atev, G. Miller, and N. P. Papanikolopoulos. 'Clustering of Vehicle Trajectories'. In: *IEEE Transactions on Intelligent Transportation Systems* 11.3 (Sept. 2010), pp. 647–657. DOI: 10.1109/TITS.2010.2048101 (cited on page 40).

[52] H. Jeung, H. T. Shen, and X. Zhou. 'Convoy Queries in Spatio-Temporal Databases'. In: *2008 IEEE 24th International Conference on Data Engineering*. Apr. 2008, pp. 1457–1459. DOI: 10.1109/ICDE.2008.4497588 (cited on page 40).

[53] Yan Liu, Yun Li, and Yun-Hao Yuan. 'A Complete Canonical Correlation Analysis for Multiview Learning'. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 3254–3258 (cited on page 40).

[54] Heysem Kaya et al. 'CCA based feature selection with application to continuous depression recognition from acoustic speech features'. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 3729–3733 (cited on page 40).

[55] W. Wang et al. 'Unsupervised learning of acoustic features via deep canonical correlation analysis'. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2015, pp. 4590–4594. DOI: 10.1109/ICASSP.2015.7178840 (cited on page 40).

[56] Magnus Borga and Hans Knutsson. 'A canonical correlation approach to blind source separation'. In: *Report LiU-IMT-EX-0062 Department of Biomedical Engineering, Linkping University* (2001) (cited on page 41).

[57] Alain de Cheveigné et al. 'Multiway canonical correlation analysis of brain data'. In: *NeuroImage* 186 (2019), pp. 728–740. DOI: https://doi.org/10.1016/j.neuroimage.2018.11.026 (cited on page 41).

[58] Natalia Y Bilenko and Jack L Gallant. 'Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging'. In: *Frontiers in neuroinformatics* 10 (2016), p. 49 (cited on pages 47, 66).

[59] Viivi Uurtio et al. 'A tutorial on canonical correlation methods'. In: *ACM Computing Surveys (CSUR)* 50.6 (2017), pp. 1–33 (cited on page 52).

[60] Parabattina Bhagath and Pradip K. Das. 'Phoneme Boundary Analysis Using Graphs'. In: *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)* (2019), pp. 1764–1768 (cited on pages 55, 66).

[61] A. Sandryhaila and J. M. F. Moura. 'Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure'. In: *IEEE Signal Processing Magazine* 31.5 (Sept. 2014), pp. 80–90. DOI: 10.1109/MSP.2014.2329213 (cited on pages 55, 56).

[62] Rainer Dahlhaus and Michael Eichler. '1 Causality and graphical models in time series analysis'. In: 2002 (cited on page 56).

[63] James Sharpnack, Aarti Singh, and Alessandro Rinaldo. 'Changepoint Detection over Graphs with the Spectral Scan Statistic'. In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Carlos M. Carvalho and Pradeep Ravikumar. Vol. 31. Proceedings of Machine Learning Research. Scottsdale, Arizona, USA: PMLR, 29 Apr–01 May 2013, pp. 545–553 (cited on page 56).

[64] Hao Chen, Nancy Zhang, et al. 'Graph-based change-point detection'. In: *The Annals of Statistics* 43.1 (2015), pp. 139–176 (cited on page 56).

[65] Xi He et al. ‘Sequential Graph Scanning Statistic for Change-point Detection’. In: Oct. 2018, pp. 1317–1321. DOI: 10.1109/ACSSC.2018.8645505 (cited on page 56).

[66] Hao Chen et al. ‘Sequential change-point detection based on nearest neighbors’. In: *The Annals of Statistics* 47.3 (2019), pp. 1381–1407 (cited on page 56).

[67] Y. Chen et al. ‘Change-Point Detection of Gaussian Graph Signals with Partial Information’. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2018, pp. 3934–3938. DOI: 10.1109/ICASSP.2018.8461397 (cited on page 56).

[68] Xinbo Gao et al. ‘A survey of graph edit distance’. In: *Pattern Analysis and Applications* 13.1 (Feb. 2010), pp. 113–129. DOI: 10.1007/s10044-008-0141-y (cited on page 60).

[69] Kaspar Riesen, Michel Neuhaus, and Horst Bunke. ‘Bipartite Graph Matching for Computing the Edit Distance of Graphs’. In: *Graph-Based Representations in Pattern Recognition*. Ed. by Francisco Escolano and Mario Vento. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–12 (cited on pages 60, 61).

[70] Kaspar Riesen and Horst Bunke. ‘Approximate graph edit distance computation by means of bipartite graph matching’. In: *Image and Vision Computing* 27.7 (2009). 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007), pp. 950–959. DOI: 10.1016/j.imavis.2008.04.004 (cited on page 60).

[71] J. Munkres. ‘Algorithms for the Assignment and Transportation Problems’. In: *Journal of the Society for Industrial and Applied Mathematics* 5.1 (1957), pp. 32–38. DOI: 10.1137/0105003 (cited on page 61).

[72] N. Meghanathan. ‘Spectral Radius as a Measure of Variation in Node Degree for Complex Network Graphs’. In: *2014 7th International Conference on u- and e- Service, Science and Technology*. Dec. 2014, pp. 30–33. DOI: 10.1109/UNESST.2014.8 (cited on page 62).

[73] C.C. Paige. ‘Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem’. In: *Linear Algebra and its Applications* 34 (1980), pp. 235–258. DOI: https://doi.org/10.1016/0024-3795(80)90167-6 (cited on page 62).

[74] Michaël Defferrard et al. *PyGSP: Graph Signal Processing in Python*. DOI: 10.5281/zenodo.1003157. URL: https://github.com/epfl-lts2/pygsp/ (cited on page 62).

[75] Cornelius Lanczos. ‘An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators1’. In: *Journal of Research of the National Bureau of Standards* 45.4 (1950) (cited on page 62).

[76] Daniel A. Schult. ‘Exploring network structure, dynamics, and function using NetworkX’. In: *In Proceedings of the 7th Python in Science Conference (SciPy*. 2008, pp. 11–15 (cited on page 66).

[77] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: http://www.scipy.org/ (cited on page 66).

[78] Clifford A. Pickover and Al Khorasani. ‘Fractal characterization of speech waveform graphs’. In: *Computers & Graphics* 10.1 (1986), pp. 51–61. DOI: 10.1016/0097-8493(86)90068-3 (cited on pages 71, 72).

[79] Francesco Mulargia, Guido Gonzato, and Warner Marzocchi. 'Practical application of fractal analysis: problems and solutions'. In: *Geophysical Journal International* 132.2 (Feb. 1998), pp. 275–282. DOI: 10.1046/j.1365-246x.1998.00461.x (cited on page 72).

[80] T. R. Senevirathne, E. L. J. Bohez, and J. A. Van Winden. 'Amplitude scale method: new and efficient approach to measure fractal dimension of speech waveforms'. In: *Electronics Letters* 28.4 (Feb. 1992), pp. 420–422. DOI: 10.1049/el:19920264 (cited on page 73).

[81] Erik L.J Bohez and T.R Senevirathne. 'Speech recognition using fractals'. In: *Pattern Recognition* 34.11 (2001), pp. 2227–2243. DOI: 10.1016/S0031-3203(00)00137-0 (cited on page 73).

[82] N. Gache, P. Flandrin, and D. Garreau. 'Fractal dimension estimators for fractional Brownian motions'. In: *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*. Apr. 1991, 3557–3560 vol.5. DOI: 10.1109/ICASSP.1991.150243 (cited on page 73).

[83] L. M. Kaplan and C. -. J. Kuo. 'An improved algorithm for fractal estimation from noisy measurements'. In: *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 4. Apr. 1993, 89–92 vol.4. DOI: 10.1109/ICASSP.1993.319601 (cited on page 73).

[84] Ivan Michieli and B Medved Rogina. 'Extracting self-affine (fractal) features from physiologic signals'. In: *2007 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*. IEEE. 2007, pp. 57–60 (cited on page 73).

[85] Jean Serra and Luc Vincent. 'An overview of morphological filtering'. In: *Circuits, Systems and Signal Processing* 11.1 (Mar. 1992), pp. 47–108. DOI: 10.1007/BF01189221 (cited on pages 73, 80).

[86] Henk J. A. M. Heijmans. 'Mathematical Morphology: A Modern Approach in Image Processing Based on Algebra and Geometry'. In: *Siam Review - SIAM REV* 37 (Mar. 1995), pp. 1–36. DOI: 10.1137/1037001 (cited on page 73).

[87] Petros Maragos and Alexandros Potamianos. 'Fractal dimensions of speech sounds: Computation and application to automatic speech recognition'. In: *The Journal of the Acoustical Society of America* 105.3 (1999), pp. 1925–1932. DOI: 10.1121/1.426738 (cited on page 74).

[88] R. Steinberg and D. O'Shaughnessy. 'Segmentation of a speech spectrogram using mathematical morphology'. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. Mar. 2008, pp. 1637–1640. DOI: 10.1109/ICASSP.2008.4517940 (cited on page 74).

[89] J. Franke et al. 'Phoneme Boundary Detection using Deep Bidirectional LSTMs'. In: *Speech Communication; 12. ITG Symposium*. Oct. 2016, pp. 1–5 (cited on page 83).

[90] A. Rendel et al. 'Towards automatic phonetic segmentation for TTS'. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2012, pp. 4533–4536. DOI: 10.1109/ICASSP.2012.6288926 (cited on page 83).

[91] Archana Balyan, S. S. Agrawal, and Amita Dev. 'Automatic phonetic segmentation of Hindi speech using hidden Markov model'. In: *AI & SOCIETY* 27.4 (Nov. 2012), pp. 543–549. DOI: 10.1007/s00146-012-0386-2 (cited on page 83).